

Tipo	Operação de API	Cota (por conta de parceiro)
	GetAwsOpportunitySummary	
	ListOpportunities	
	ListSolutions	
	GetEngagementInvitation	
	ListEngagementInvitations	
Prospecção, leitura, ações	GetProspectingFromEngagementTask	100 por segundo
Prospecção, leitura, ações	ListProspectingFromEngagementTasks	50 por segundo
Escrever ações	CreateOpportunity	1 por segundo; 10.000 por 24 horas
	UpdateOpportunity	
	AssociateOpportunity	
	DisassociateOpportunity	
	RejectEngagementInvitation	
	AssignOpportunity	
	StartEngagementFromOpportunityTask	
	StartEngagementByAcceptingInvitationTask	
Prospecção, escrita, ações	StartProspectingFromEngagementTask	2 por segundo
Engajamento, escreva ações	CreateEngagement	15 por segundo

Cotas de associação por oportunidade

Entidade relacionada	Quota
AWS produtos	20 por oportunidade
Soluções para parceiros	10 por oportunidade
AWS Marketplace soluções	10 por oportunidade
AWS Marketplace produtos	10 por oportunidade
AWS Marketplace ofertas privadas	1 por oportunidade

Entendendo e gerenciando cotas

Limitação de intervalo

Quando um limite de taxa de API for atingido, o serviço responderá com um `ThrottlingException`. Para lidar melhor com a limitação de taxa, a AWS recomenda implementar estratégias exponenciais de recuo e repetição em seu aplicativo.

Janela de tempo para cotas

As cotas diárias são redefinidas em um período contínuo de 24 horas. Suas solicitações seriam limitadas, por exemplo, se você tivesse realizado 10.000 ações de gravação nas últimas 24 horas e estivesse tentando realizar a 10.001ª solicitação. Certifique-se de que os padrões de uso do seu aplicativo levem isso em consideração para evitar a limitação não intencional.

Solicitar um aumento de cota

Se as cotas padrão não atenderem aos seus requisitos, você poderá solicitar um aumento de cota na página [Service Quotas](#). O console Service Quotas é uma interface baseada em navegador que você pode usar para visualizar e gerenciar suas cotas de serviço. Você pode acessar as Cotas de Serviço de qualquer Console de gerenciamento da AWS página escolhendo-a na barra de navegação superior ou pesquisando por Cotas de Serviço no Console de gerenciamento da AWS.

Cotas para o AWS API de benefícios do Partner Central

A API de benefícios do AWS Partner Central tem as seguintes cotas.

Cotas de solicitações

Cotas de solicitações

Operações de API	Cota (por AWS conta)
ListBenefits	20 por segundo
GetBenefit	20 por segundo
AmendBenefitApplication	10 por segundo
CancelBenefitApplication	10 por segundo
CreateBenefitApplication	10 por segundo
GetBenefitApplication	20 por segundo
ListBenefitApplications	30 por segundo
RecallBenefitApplication	10 por segundo
SubmitBenefitApplication	10 por segundo
UpdateBenefitApplication	10 por segundo
GetBenefitAllocation	20 por segundo
ListBenefitAllocations	20 por segundo
AssociateBenefitApplicationResource	10 por segundo
DisassociateBenefitApplicationResource	10 por segundo

Cotas adicionais

Cotas adicionais

Nome de exibição	Catálogo	Description	Valor padrão
Aplicativos de benefícios	AWS	O número máximo de aplicativos de	10.000

Nome de exibição	Catálogo	Description	Valor padrão
		benefícios que você pode criar no AWS catálogo	
Aplicativos de benefícios	Sandbox	O número máximo de aplicativos de benefícios que você pode criar no catálogo do Sandbox	10.000

Entendendo e gerenciando cotas

Limitação de intervalo

Quando um limite de taxa de API for atingido, o serviço responderá com um `ThrottlingException`. Para lidar melhor com a limitação de taxa, a AWS recomenda implementar estratégias exponenciais de recuo e repetição em seu aplicativo.

Solicitar um aumento de cota

Se as cotas padrão não atenderem aos seus requisitos, você poderá solicitar um aumento de cota na página [Service Quotas](#). O console Service Quotas é uma interface baseada em navegador que você pode usar para visualizar e gerenciar suas cotas de serviço. Você pode acessar as Cotas de Serviço de qualquer Console de gerenciamento da AWS página escolhendo-a na barra de navegação superior ou pesquisando por Cotas de Serviço no Console de gerenciamento da AWS.

Cotas para o AWS API do Partner Central Channel

A API do AWS Partner Central Channel tem as seguintes cotas.

Cotas de solicitações

Cotas de solicitações

Operações de API	Cota (por AWS conta)
CreateProgramManagementAccount	3 por segundo

Operações de API	Cota (por AWS conta)
UpdateProgramManagementAccount	3 por segundo
ListProgramManagementAccounts	5 por segundo
DeleteProgramManagementAccount	3 por segundo
CreateChannelHandshake	3 por segundo
ListChannelHandshakes	5 por segundo
AcceptChannelHandshake	3 por segundo
RejectChannelHandshake	3 por segundo
CancelChannelHandshake	3 por segundo
CreateRelationship	3 por segundo
GetRelationship	5 por segundo
ListRelationships	5 por segundo
UpdateRelationship	3 por segundo
DeleteRelationship	3 por segundo

Cotas adicionais

Cotas adicionais

Nome de exibição	Catálogo	Description	Valor padrão
Contas de gerenciamento do programa	AWS	O número máximo de contas de gerenciamento de programas que você pode criar no AWS catálogo	50

Nome de exibição	Catálogo	Description	Valor padrão
Relacionamentos por conta de gerenciamento do programa	AWS	O número máximo de relacionamentos que você pode estabelecer por conta de gerenciamento do programa no AWS catálogo	1.000
Apertos de mão da conta de gerenciamento do programa por conta	AWS	O número máximo de apertos de mão ativos da conta de gerenciamento de programas no catálogo AWS	100
Início do período de serviço: apertos de mão por conta	AWS	O número máximo de apertos de mão ativos para estabelecer períodos de serviço no catálogo AWS	200
Revogue os apertos de mão do período de serviço por conta	AWS	O número máximo de apertos de mão ativos para revogar períodos de serviço no catálogo AWS	200
Taxa de apertos de mão da conta de gerenciamento de programas por dia	AWS	O número máximo de apertos de mão da conta de gerenciamento do programa que você pode enviar para a mesma AWS conta por dia no catálogo AWS	5

Nome de exibição	Catálogo	Description	Valor padrão
Taxa de apertos de mão do período de início do serviço por dia	AWS	O número máximo de apertos de mão do período de início do serviço que você pode enviar para a mesma AWS conta por dia no catálogo AWS	5
Taxa de apertos de mão do período de revogação do serviço por dia	AWS	O número máximo de apertos de mão do período de revogação do serviço que você pode enviar para a mesma AWS conta por dia no catálogo AWS	5
Contas de gerenciamento do programa	Sandbox	O número máximo de contas de gerenciamento de programas que você pode criar no catálogo do Sandbox	50
Relacionamentos por conta de gerenciamento do programa	Sandbox	O número máximo de relacionamentos que você pode estabelecer por conta de gerenciamento do programa no catálogo do Sandbox	1.000

Nome de exibição	Catálogo	Description	Valor padrão
Apertos de mão da conta de gerenciamento do programa por conta	Sandbox	O número máximo de apertos de mão ativos da conta de gerenciamento de programas no catálogo do Sandbox	100
Início do período de serviço: apertos de mão por conta	Sandbox	O número máximo de apertos de mão ativos para estabelecer períodos de serviço no catálogo do Sandbox	200
Revogue os apertos de mão do período de serviço por conta	Sandbox	O número máximo de apertos de mão ativos para revogar períodos de serviço no catálogo do Sandbox	200
Taxa de apertos de mão da conta de gerenciamento de programas por dia	Sandbox	O número máximo de apertos de mão da conta de gerenciamento do programa que você pode enviar para a mesma AWS conta por dia no catálogo do Sandbox	5

Nome de exibição	Catálogo	Description	Valor padrão
Taxa de apertos de mão do período de início do serviço por dia	Sandbox	O número máximo de apertos de mão do período de início do serviço que você pode enviar para a mesma AWS conta por dia no catálogo do Sandbox	5
Taxa de apertos de mão do período de revogação do serviço por dia	Sandbox	O número máximo de apertos de mão do período de revogação do serviço que você pode enviar para a mesma AWS conta por dia no catálogo do Sandbox	5

Entendendo e gerenciando cotas

Limitação de intervalo

Quando um limite de taxa de API for atingido, o serviço responderá com um `ThrottlingException`. Para lidar melhor com a limitação de taxa, a AWS recomenda implementar estratégias exponenciais de recuo e repetição em seu aplicativo.

Solicitar um aumento de cota

Se as cotas padrão não atenderem aos seus requisitos, você poderá solicitar um aumento de cota na página [Service Quotas](#). O console Service Quotas é uma interface baseada em navegador que você pode usar para visualizar e gerenciar suas cotas de serviço. Você pode acessar as Cotas de Serviço de qualquer Console de gerenciamento da AWS página escolhendo-a na barra de navegação superior ou pesquisando por Cotas de Serviço no Console de gerenciamento da AWS.

Cotas para o AWS API de medição de receita do Partner Central

AWS A API de medição de receita do Partner Central impõe cotas para garantir o uso justo e proteger o serviço contra uso indevido. Abaixo estão as cotas detalhadas para várias operações de API.

Cotas de operação da API

Tipo	Operação de API	Cota (por conta de parceiro)
Leia as ações	GetRevenueAttribution	50 por segundo
	GetRevenueAttributionAllocation	
	GetRevenueAttributionAllocationsTask	
	GetMarketplaceRevenueShare	
	GetMarketplaceRevenueShareAllocation	
Listar ações	ListRevenueAttributions	10 por segundo
	ListRevenueAttributionAllocations	
	ListMarketplaceRevenueShares	
	ListMarketplaceRevenueShareAllocations	
	ListTagsForResource	
Escrever ações	CreateRevenueAttribution	2 por segundo
	UpdateRevenueAttribution	

Tipo	Operação de API	Cota (por conta de parceiro)
	CreateMarketplaceRevenueShare	
	CreateMarketplaceRevenueShareAllocation	
	UpdateMarketplaceRevenueShareAllocation	
	TagResource	
	UntagResource	
Ações de gravação assíncrona	StartRevenueAttributionAllocationsTask	1 por segundo

Entendendo e gerenciando cotas

Limitação de intervalo

Quando um limite de taxa de API for atingido, o serviço responderá com um `ThrottlingException`. Para lidar melhor com a limitação de taxa, AWS recomenda implementar estratégias exponenciais de recuo e repetição em seu aplicativo.

Solicitar um aumento de cota

Se as cotas padrão não atenderem aos seus requisitos, você poderá solicitar um aumento de cota na página [Service Quotas](#). O console Service Quotas é uma interface baseada em navegador que você pode usar para visualizar e gerenciar suas cotas de serviço. Você pode acessar as Cotas de Serviço de qualquer Console de gerenciamento da AWS página escolhendo-a na barra de navegação superior ou pesquisando por Cotas de Serviço no Console de gerenciamento da AWS.

Registro para o AWS API do Partner Central

As seções a seguir fornecem informações sobre registro em log.

Tópicos

- [Registrando o AWS API de conta do Partner Central](#)
- [Registrando o AWS API de vendas da Central de Parceiros](#)
- [Registrando o AWS API de benefícios do Partner Central](#)
- [Registrando o AWS API do Partner Central Channel](#)
- [Registrando o AWS API de atribuição de receita do Partner Central](#)

Registrando o AWS API de conta do Partner Central

[AWS CloudTrail](#) é um serviço que permite governança, conformidade, auditoria operacional e auditoria de risco de seu. Conta da AWS Com AWS CloudTrail, você pode registrar, monitorar continuamente e reter as atividades da conta relacionadas às ações em toda a sua AWS infraestrutura. A atividade da API da conta do Partner Central é registrada como eventos em CloudTrail. Você pode criar uma trilha, uma configuração que permite a entrega de eventos como arquivos de log para um bucket do Amazon S3.

Visão geral do

A API de conta do AWS Partner Central está integrada com AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, função ou AWS serviço no AWS Partner Central. CloudTrail captura todas as chamadas de API para a API da conta do AWS Partner Central como eventos. As chamadas capturadas incluem chamadas do AWS Partner Central e de chamadas de código para as operações da API da conta do AWS Partner Central.

Se você criar uma trilha, poderá habilitar a entrega contínua de CloudTrail eventos para um bucket do Amazon S3, incluindo eventos para a API de contas do AWS Partner Central. Se você não configurar uma trilha, ainda poderá ver os eventos mais recentes no CloudTrail console no Histórico de eventos.

Usando as informações coletadas por CloudTrail, você pode determinar a solicitação feita à API da Conta do AWS Partner Central, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Noções básicas AWS Entradas do arquivo de log da Partner Central Account API

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log para um bucket do Amazon S3. Quando sua trilha acompanha os eventos da API da Conta do AWS Partner Central, CloudTrail processa os eventos como arquivos de log em todas as regiões. Cada arquivo de log pode conter um ou mais eventos.

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra a `CreatePartner` ação na API de conta do AWS Partner Central:

```
{
  "eventVersion": "1.11",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:user/CloudTrailTestUser",
    "accountId": "123456789012",
    "accessKeyId": "ABCDEFGHijklmnop1234",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROEXAMPLE52AGFT725JGDZ",
        "arn": "arn:aws:iam::123456789012:role/ExampleRole",
        "accountId": "123456789012",
        "userName": "ExampleRole"
      },
      "attributes": {
        "creationDate": "2025-11-07T19:45:28Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2025-11-07T19:45:58Z",
  "eventSource": "partnercentral-account.amazonaws.com",
  "eventName": "CreatePartner",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "PostmanRuntime/7.18.0",
  "requestParameters": {
    "catalog": "AWS",
    "clientToken": "abcdef12-3456-7890-bcde-f123456789ab",
    "legalName": "ExampleLegalName",
    "primarySolutionType": "VALUE_ADDED_RESALE_AWS_SERVICES",
    "allianceLeadContact": {
      "firstName": "ExampleFirstName",
      "lastName": "ExampleLastName",
      "email": "example@domain.com",
      "businessTitle": "ExampleBusinessTitle"
    }
  },
  "emailVerificationCode": "ExampleVerificationCode",
```

```

    "tags": [
      {
        "key": "office",
        "value": "1"
      }
    ],
  },
  "responseElements": {
    "catalog": "AWS",
    "arn": "arn:aws:partnercentral:us-east-1:123456789012:catalog/AWS/partner/
EXAMPLE_PARTNER_ID",
    "id": "EXAMPLE_PARTNER_ID",
    "legalName": "ExampleLegalName",
    "createdAt": "2025-11-07T19:45:57.867007329Z",
    "profile": {
      "primarySolutionType": "VALUE_ADDED_RESALE_AWS_SERVICES"
    },
    "allianceLeadContact": {
      "firstName": "ExampleFirstName",
      "lastName": "ExampleLastName",
      "email": "example@domain.com",
      "businessTitle": "ExampleBusinessTitle"
    }
  },
  "requestID": "12345678-1234-5678-9abc-def012345678",
  "eventID": "87654321-4321-8765-cba9-fed098765432",
  "readOnly": false,
  "resources": [
    {
      "accountId": "123456789012",
      "type": "AWS::PartnerCentral::Partner",
      "ARN": "arn:aws:partnercentral:us-east-1:123456789012:catalog/AWS/partner/
EXAMPLE_PARTNER_ID"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management",
  "tlsDetails": {
    "tlsVersion": "TLSv1.3",
    "cipherSuite": "TLS_AES_128_GCM_SHA256",
    "clientProvidedHostHeader": "partnercentral-account.partner.us-east-1.api.aws"
  }
}

```

```
}
```

Neste exemplo, a `CreatePartner` ação foi chamada pelo usuário do IAM chamado `CloudTrailTestUser`. A solicitação foi feita em 7 de novembro de 2025 às 19:45:58 UTC. A solicitação criou um novo parceiro com ID `EXAMPLE_PARTNER_ID` para o catálogo da AWS com o nome legal `"ExampleLegalName"`.

Campos em AWS Entradas do arquivo de log da Partner Central Account API

Cada entrada em um arquivo de CloudTrail log contém informações sobre quem fez uma solicitação, os recursos utilizados na solicitação e os elementos de resposta retornados pela API de conta do AWS Partner Central. A lista de campos em uma entrada de registro, como `eventVersion`, `userIdentity`, `eventTime`, fornece informações detalhadas sobre a ação. Por exemplo, o `sourceIPAddress` campo mostra o endereço IP do qual a solicitação foi feita.

Registrando o AWS API de vendas da Central de Parceiros

[AWS CloudTrail](#) é um serviço que permite governança, conformidade, auditoria operacional e auditoria de risco de seu. Conta da AWS Com AWS CloudTrail, você pode registrar, monitorar continuamente e reter as atividades da conta relacionadas às ações em toda a sua AWS infraestrutura. A atividade da API do Partner Central é registrada como eventos em CloudTrail. Você pode criar uma trilha, uma configuração que permite a entrega de eventos como arquivos de log para um bucket do Amazon S3.

Visão geral do

A API do AWS Partner Central é integrada com AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, função ou AWS serviço no AWS Partner Central. CloudTrail captura todas as chamadas de API para o AWS Partner Central como eventos. As chamadas capturadas incluem chamadas do AWS Partner Central e de chamadas de código para as operações da API do AWS Partner Central.

Se você criar uma trilha, poderá habilitar a entrega contínua de CloudTrail eventos para um bucket do Amazon S3, incluindo eventos para o AWS Partner Central. Se você não configurar uma trilha, ainda poderá ver os eventos mais recentes no CloudTrail console no Histórico de eventos.

Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita ao AWS Partner Central, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Noções básicas AWS Entradas do arquivo de registro da API de vendas do Partner Central

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log para um bucket do Amazon S3. Quando sua trilha acompanha os eventos do AWS Partner Central, CloudTrail processa os eventos como arquivos de log em todas as regiões. Cada arquivo de log pode conter um ou mais eventos.

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra a `ListOpportunities` ação no AWS Partner Central:

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "ABCDEFGHJKLMNOP12345",
    "arn": "arn:aws:iam::123456789010:user/CloudTrailTestUser",
    "accountId": "123456789010",
    "accessKeyId": "ABCDEFGHJKLMNOP1234",
    "userName": "CloudTrailTestUser"
  },
  "eventTime": "2023-10-17T21:49:23Z",
  "eventSource": "partnercentral-selling.amazonaws.com",
  "eventName": "ListOpportunities",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "PostmanRuntime/7.18.0",
  "requestParameters": {
    "MaxResults": 20
  },
  "responseElements": null,
  "requestID": "fEXAMPLE-cb3e-4e21-86fd-6b3EXAMPLEd1",
  "eventID": "7EXAMPLE-97d6-4139-91e3-01aEXAMPLE48",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "recipientAccountId": "123456789010"
}
```

Neste exemplo, a `ListOpportunities` ação foi chamada pelo usuário do IAM chamado `CloudTrailTestUser`. A ação foi chamada no Região da AWS `us-east-1` e a solicitação foi feita em 17 de outubro de 2023 às 21:49:23 UTC.

Campos em AWS Entradas do arquivo de registro da API de vendas do Partner Central

Cada entrada em um arquivo de CloudTrail log contém informações sobre quem fez uma solicitação, os recursos utilizados na solicitação e os elementos de resposta retornados pelo AWS Partner Central. A lista de campos em uma entrada de registro, como `eventVersion`, `userIdentity`, `eventTime`, fornece informações detalhadas sobre a ação. Por exemplo, o `sourceIPAddress` campo mostra o endereço IP a partir do qual a solicitação foi feita.

Registrando o AWS API de benefícios do Partner Central

[AWS CloudTrail](#) é um serviço que permite a governança, a conformidade, a auditoria operacional e a auditoria de risco do seu. Conta da AWS Com AWS CloudTrail, você pode registrar, monitorar continuamente e reter as atividades da conta relacionadas às ações em toda a sua AWS infraestrutura. A atividade da API de benefícios do Partner Central é registrada como eventos em CloudTrail. Você pode criar uma trilha, uma configuração que permite a entrega de eventos como arquivos de log para um bucket do Amazon S3.

Visão geral do

A API de benefícios do AWS Partner Central está integrada com AWS CloudTrail um serviço que fornece um registro das ações realizadas por um usuário, função ou AWS serviço no AWS Partner Central. CloudTrail captura todas as chamadas de API para a API de benefícios do AWS Partner Central como eventos. As chamadas capturadas incluem chamadas do AWS Partner Central e de chamadas de código para as operações da API de benefícios do AWS Partner Central.

Se você criar uma trilha, poderá habilitar a entrega contínua de CloudTrail eventos para um bucket do Amazon S3, incluindo eventos para a API de benefícios do AWS Partner Central. Se você não configurar uma trilha, ainda poderá ver os eventos mais recentes no CloudTrail console no Histórico de eventos.

Usando as informações coletadas por CloudTrail, você pode determinar a solicitação feita à API de benefícios do AWS Partner Central, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Noções básicas AWS Entradas do arquivo de log da API de benefícios do Partner Central

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log para um bucket do Amazon S3. Quando sua trilha acompanha os eventos da API de benefícios do AWS Partner Central, CloudTrail processa os eventos como arquivos de log em todas as regiões. Cada arquivo de log pode conter um ou mais eventos.

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra a `ListBenefitApplications` ação na API de benefícios do AWS Partner Central:

```
{
  "eventVersion": "1.11",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:user/CloudTrailTestUser",
    "accountId": "123456789012",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:role/TestRole",
        "accountId": "123456789012",
        "userName": "TestRole"
      },
      "attributes": {
        "creationDate": "2025-10-21T03:08:23Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2025-10-21T03:10:59Z",
  "eventSource": "partnercentral-benefits.amazonaws.com",
  "eventName": "ListBenefitApplications",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "python-requests/2.32.4",
  "requestParameters": {
    "catalog": "AWS"
  },
}
```

```
"responseElements": null,
"requestID": "12345678-1234-5678-9abc-def012345678",
"eventID": "87654321-4321-8765-cba9-fed098765432",
"readOnly": true,
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "123456789012",
"eventCategory": "Management"
}
```

Neste exemplo, a `ListBenefitApplications` ação foi chamada pela usuária do IAM chamada Alice. A solicitação foi feita em 21 de outubro de 2025 às 03:10:59 UTC. A solicitação listou os aplicativos de benefícios para o catálogo da AWS.

Campos em AWS Entradas do arquivo de log da API de benefícios do Partner Central

Cada entrada em um arquivo de CloudTrail log contém informações sobre quem fez uma solicitação, os recursos utilizados na solicitação e os elementos de resposta retornados pela API de benefícios do AWS Partner Central. A lista de campos em uma entrada de registro, como `eventVersion`, `userIdentity`, `eventTime`, fornece informações detalhadas sobre a ação. Por exemplo, o `sourceIPAddress` campo mostra o endereço IP a partir do qual a solicitação foi feita.

Registrando o AWS API do Partner Central Channel

[AWS CloudTrail](#) é um serviço que permite a governança, a conformidade, a auditoria operacional e a auditoria de risco do seu. Conta da AWS Com AWS CloudTrail, você pode registrar, monitorar continuamente e reter as atividades da conta relacionadas às ações em toda a sua AWS infraestrutura. A atividade da API do Partner Central Channel é registrada como eventos em CloudTrail. Você pode criar uma trilha, uma configuração que permite a entrega de eventos como arquivos de log para um bucket do Amazon S3.

Visão geral do

A API do AWS Partner Central Channel é integrada com AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, função ou AWS serviço no AWS Partner Central. CloudTrail captura todas as chamadas de API para a API do AWS Partner Central Channel como eventos. As chamadas capturadas incluem chamadas do AWS Partner Central e de chamadas de código para as operações da API do AWS Partner Central Channel.

Se você criar uma trilha, poderá habilitar a entrega contínua de CloudTrail eventos para um bucket do Amazon S3, incluindo eventos para a API do AWS Partner Central Channel. Se você não

configurar uma trilha, ainda poderá ver os eventos mais recentes no CloudTrail console no Histórico de eventos.

Usando as informações coletadas por CloudTrail, você pode determinar a solicitação feita à API do AWS Partner Central Channel, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Noções básicas AWS Entradas do arquivo de log da API Partner Central Channel

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log para um bucket do Amazon S3. Quando sua trilha acompanha os eventos da API do AWS Partner Central Channel, CloudTrail processa os eventos como arquivos de log em todas as regiões. Cada arquivo de log pode conter um ou mais eventos.

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra a `CreateProgramManagementAccount` ação na API do AWS Partner Central Channel:

```
{
  "eventVersion": "1.11",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROEXAMPLE52AGFT725JGDZ:example-user-Isengard",
    "arn": "arn:aws:sts::123456789012:assumed-role/Admin/example-user-Isengard",
    "accountId": "123456789012",
    "accessKeyId": "ASIAEXAMPLE52AGL6IXQLZ5",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROEXAMPLE52AGFT725JGDZ",
        "arn": "arn:aws:iam::123456789012:role/ExampleRole",
        "accountId": "123456789012",
        "userName": "ExampleRole"
      },
      "attributes": {
        "creationDate": "2025-10-21T17:06:47Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2025-10-21T17:07:26Z",
  "eventSource": "partnercentral-channel.amazonaws.com",
  "eventName": "CreateProgramManagementAccount",
```

```

    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "PostmanRuntime/7.18.0",
    "requestParameters": {
      "catalog": "AWS",
      "program": "SOLUTION_PROVIDER",
      "displayName": "ExampleDisplayName",
      "accountId": "987654321098",
      "clientToken": "abcdef12-3456-7890-bcde-f123456789ab"
    },
    "responseElements": {
      "programManagementAccountDetail": {
        "id": "pma-example123456789",
        "arn": "arn:aws:partnercentral:us-east-1:123456789012:catalog/AWS/program-
management-account/pma-example123456789"
      }
    },
    "requestID": "12345678-1234-5678-9abc-def012345678",
    "eventID": "87654321-4321-8765-cba9-fed098765432",
    "readOnly": false,
    "resources": [
      {
        "accountId": "123456789012",
        "type": "AWS::PartnerCentralChannel::ProgramManagementAccount",
        "ARN": "arn:aws:partnercentral:us-east-1:123456789012:catalog/AWS/program-
management-account/pma-example123456789"
      }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management",
    "tlsDetails": {
      "clientProvidedHostHeader": "partnercentral-channel.global.api.aws"
    }
  }
}

```

Neste exemplo, a `CreateProgramManagementAccount` ação foi chamada pela função do IAM nomeada por `ExampleRole` meio de uma sessão de função assumida. A solicitação foi feita em 21 de outubro de 2025 às 17:07:26 UTC. A solicitação criou uma nova conta de gerenciamento do programa com ID `pma-example123456789` para o programa `Solution Provider`.

Campos em AWS Entradas do arquivo de log da API Partner Central Channel

Cada entrada em um arquivo de CloudTrail log contém informações sobre quem fez uma solicitação, os recursos utilizados na solicitação e os elementos de resposta retornados pela API do AWS Partner Central Channel. A lista de campos em uma entrada de registro, como `eventVersion`, `userIdentity`, `eventTime`, fornece informações detalhadas sobre a ação. Por exemplo, o `sourceIPAddress` campo mostra o endereço IP a partir do qual a solicitação foi feita.

Registrando o AWS API de atribuição de receita do Partner Central

[AWS CloudTrail](#) é um serviço que permite a governança, a conformidade, a auditoria operacional e a auditoria de risco do seu. Conta da AWS Com AWS CloudTrail, você pode registrar, monitorar continuamente e reter as atividades da conta relacionadas às ações em toda a sua AWS infraestrutura. A atividade da API de medição de receita do Partner Central é registrada como eventos em CloudTrail. Você pode criar uma trilha, uma configuração que permite a entrega de eventos como arquivos de log para um bucket do Amazon S3.

Visão geral do

A API de Medição de Receita do AWS Partner Central está integrada com AWS CloudTrail um serviço que fornece um registro das ações realizadas por um usuário, função ou AWS serviço no AWS Partner Central. CloudTrail captura todas as chamadas de API para o AWS Partner Central Revenue Attribution como eventos. As chamadas capturadas incluem chamadas do console do AWS Partner Central e de chamadas de código para as operações da API de Medição de Receita do AWS Partner Central.

Se você criar uma trilha, poderá habilitar a entrega contínua de CloudTrail eventos para um bucket do Amazon S3, incluindo eventos para o AWS Partner Central Revenue Measurement. Se você não configurar uma trilha, ainda poderá ver os eventos mais recentes no CloudTrail console no Histórico de eventos.

Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita à AWS Partner Central Revenue Measurement, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

As seguintes ações da API de Medição de Receita do AWS Partner Central estão registradas: CloudTrail

- `CreateRevenueAttribution`

- UpdateRevenueAttribution
- GetRevenueAttribution
- ListRevenueAttributions
- TagResource
- UntagResource
- ListTagsForResource
- CreateMarketplaceRevenueShare
- GetMarketplaceRevenueShare
- ListMarketplaceRevenueShares
- CreateMarketplaceRevenueShareAllocation
- GetMarketplaceRevenueShareAllocation
- UpdateMarketplaceRevenueShareAllocation
- ListMarketplaceRevenueShareAllocations
- StartRevenueAttributionAllocationsTask
- GetRevenueAttributionAllocationsTask
- ListRevenueAttributionAllocations
- GetRevenueAttributionAllocation

Noções básicas AWS Entradas do arquivo de registro da API de medição de receita do Partner Central

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log para um bucket do Amazon S3. Quando sua trilha acompanha os eventos de medição de receita do AWS Partner Central, CloudTrail processa os eventos como arquivos de log em todas as regiões. Cada arquivo de log pode conter um ou mais eventos.

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra a CreateRevenueAttribution ação na Atribuição de Receita do AWS Partner Central:

```
{
  "eventVersion": "1.11",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAEXAMPLEID",
```

```

    "arn": "arn:aws:iam::123456789012:user/CloudTrailTestUser",
    "accountId": "123456789012",
    "accessKeyId": "AKIAEXAMPLEKEY",
    "userName": "CloudTrailTestUser"
  },
  "eventTime": "2026-06-02T11:53:54Z",
  "eventSource": "partnercentral-prm.amazonaws.com",
  "eventName": "CreateRevenueAttribution",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.1",
  "userAgent": "aws-sdk-java/2.20.0",
  "requestParameters": {
    "catalog": "AWS",
    "name": "my-revenue-attribution",
    "marketplaceProduct": {
      "tenancyModel": "MULTI_TENANT"
    }
  },
  "responseElements": {
    "id": "ra-0123456789abcdef0",
    "arn": "arn:aws:partnercentral:us-east-1:123456789012:catalog/AWS/revenue-attribution/ra-0123456789abcdef0",
    "name": "my-revenue-attribution",
    "revision": "1"
  },
  "requestID": "5ce2f907-3850-412a-b1a4-r012r1234567",
  "eventID": "80b39b72-ee2e-4578-8db0-01ee2e34ee56",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "recipientAccountId": "123456789012"
}

```

Neste exemplo, a `CreateRevenueAttribution` ação foi chamada pelo usuário do IAM chamado `CloudTrailTestUser`. A ação foi chamada em Região da AWS `us-east-1` e a solicitação foi feita em 2 de junho de 2026 às 11:53:54 UTC. Um novo recurso de atribuição de receita foi criado com o ID. `ra-0123456789abcdef0`

Campos em AWS Entradas do arquivo de registro da API de medição de receita do Partner Central

Cada entrada em um arquivo de CloudTrail registro contém informações sobre quem fez uma solicitação, os recursos utilizados na solicitação e os elementos de resposta retornados pela AWS

Partner Central Revenue Measurement. A lista de campos em uma entrada de registro, como `eventVersionuserIdentity`, `eventTime`, fornece informações detalhadas sobre a ação. Por exemplo, o `sourceIPAddress` campo mostra o endereço IP a partir do qual a solicitação foi feita.

Notificações para o AWS API do Partner Central

As seções a seguir fornecem informações sobre notificações.

Tópicos

- [Notificações para o AWS API de conta do Partner Central](#)
- [Notificações para o AWS API de vendas da Central de Parceiros](#)
- [Notificações para o AWS API de benefícios do Partner Central](#)
- [Notificações para o AWS API do Partner Central Channel](#)

Notificações para o AWS API de conta do Partner Central

As notificações de conexão da conta do parceiro permitem que os parceiros se mantenham informados sobre os eventos do ciclo de vida da conexão que afetam diretamente seus relacionamentos comerciais e fluxos de trabalho operacionais. Esses eventos são essenciais para que os parceiros:

- Reaja prontamente à nova colaboração
- Mantenha o conhecimento do status do portfólio de conexões
- Tome as medidas apropriadas quando as conexões forem estabelecidas ou encerradas
- Garanta a continuidade dos negócios sabendo quando os relacionamentos mudam

Tópicos

- [Preencha o pré-requisito para monitorar eventos](#)
- [Configure EventBridge a Amazon para monitorar eventos](#)
- [Saiba mais sobre eventos da API de conexões de contas](#)

Preencha o pré-requisito para monitorar eventos

Os usuários precisam das permissões apropriadas do IAM para acessar e gerenciar eventos publicados pela API de conexões da conta. Para obter mais informações sobre as ações, os recursos

e as chaves de condição disponíveis para EventBridge, consulte [Usando as condições da política do IAM na Amazon EventBridge](#) no Guia EventBridge do usuário da Amazon. Uma das chaves de condição é `events:detail-type`, que pode ser usada para definir permissões para tipos específicos de eventos.

O exemplo de política a seguir demonstra como personalizar e definir o escopo das permissões para os eventos propostos. A `AllowPutRuleForPartnercentralAccountEvents` declaração permite a criação de regras, mas somente para eventos da `aws.partnercentral-account` fonte.

Para exemplos detalhados de políticas do IAM, consulte a documentação da AWS sobre EventBridge permissões.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowPutRuleForPartnerCentralAccountEvents",
      "Effect": "Allow",
      "Action": "events:PutRule",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "events:source": "aws.partnercentral-account.connection"
        }
      }
    }
  ]
}
```

Configure EventBridge a Amazon para monitorar eventos

Para monitorar os eventos da API de conexões da conta, você cria uma EventBridge regra que corresponde aos eventos que você deseja capturar. Você pode usar o Console de gerenciamento da AWS ou os AWS SDKs para criar e gerenciar regras. As seções a seguir explicam como criar regras usando os dois métodos. Independentemente do método usado, você deve criar a regra na `us-east-1` região Leste dos EUA (Norte da Virgínia).

Console de gerenciamento da AWS configuração

Para configurar uma EventBridge regra usando o Console de gerenciamento da AWS, siga as etapas em [Criação de regras que reagem a eventos na Amazon EventBridge](#) no Guia EventBridge do

usuário da Amazon. Ao criar a regra, você deve definir o barramento de eventos como padrão e criar a regra na us-east-1 região Leste dos EUA (Norte da Virgínia).

Veja a seguir um exemplo de uma regra de evento:

```
{
  "source": ["aws.partnercentral-account"],
  "detail": {
    "catalog": ["AWS"]
  }
}
```

AWS Configuração do SDK

Você pode usar os AWS SDKs para criar e gerenciar EventBridge regras de forma programática. Para obter mais informações, consulte [PutRule](#) a Amazon EventBridge API Reference.

O exemplo a seguir usa o AWS SDK para Python (Boto3):

```
import boto3

client = boto3.client('events', region_name='us-east-1')

response = client.put_rule(
    Name='MyConnectionInvitationReceivedRule',
    EventPattern=
    '{
      "source": ["aws.partnercentral-account"],
      "detail-type": ["Partner Connection Invitation Received"],
      "detail": {"catalog": ["AWS"]}
    }',
    State='ENABLED'
)
print('Rule ARN:', response['RuleArn'])
```

Saiba mais sobre eventos da API de conexões de contas

As seções a seguir descrevem os tipos de eventos da API de conexões de conta, os cenários que os acionam e os exemplos de eventos.

Tipos de eventos

A seguir estão os tipos de eventos e seus gatilhos.

- [Convite de conexão de parceiro recebido](#): notifica o receptor de que outro parceiro deseja estabelecer uma conexão
- [Convite de conexão de parceiro aceito](#): notifica o remetente de que o convite foi aceito e que a conexão agora está ativa
- [Convite de conexão de parceiro rejeitado](#): notifica o remetente de que o convite foi recusado
- [Conexão de parceiro cancelada](#): notifica o outro participante conectado quando uma conexão ativa é encerrada
- [Convite de conexão de parceiro cancelado](#): notifica o destinatário de que o remetente retirou o convite antes que qualquer ação fosse tomada
- [Convite de conexão de parceiro expirado](#): notifica o remetente e o destinatário de que o convite expirou

Eventos de exemplo

As seções a seguir fornecem exemplos dos eventos listados anteriormente na seção anterior.

Tópicos

- [Convite de conexão de parceiro recebido](#)
- [Convite de conexão de parceiro aceito](#)
- [Convite de conexão de parceiro rejeitado](#)
- [Conexão de parceiro cancelada](#)
- [Convite de conexão de parceiro cancelado](#)
- [O convite de conexão de parceiro expirou](#)

Convite de conexão de parceiro recebido

Contexto de acionamento

Esse evento é gerado quando um convite de conexão é criado e mantido com sucesso no armazenamento de dados do PAC por meio da CreateConnectionInvitation API. O evento é enviado imediatamente após a conclusão bem-sucedida da criação do convite e persiste no armazenamento de dados do PAC, não apenas quando a chamada da API é feita.

O evento será enviado automaticamente quando um novo convite de conexão for criado. Um evento por convite criado sem eventos duplicados para o mesmo convite.

Destinatários

A AWS conta do destinatário no convite de conexão.

Tratamento esperado

Ações típicas do cliente:

- Notificação imediata: acione alertas para as equipes de negócios sobre novas oportunidades de parceria
- Automação do fluxo de trabalho: atualize automaticamente os sistemas de gerenciamento de parceiros com convites pendentes
- Apoio à decisão: encaminhe convites para os tomadores de decisão apropriados com base no tipo de conexão
- Registro de auditoria: registre o recebimento do convite para acompanhamento de conformidade e parceria
- Automação de respostas: para parceiros confiáveis, é possível aceitar automaticamente determinados tipos de convite

Padrões comuns de integração:

- Função Lambda → Atualizar o painel do portal do parceiro
- Fila SQS → Processar convites em lote para análise
- Tópico do SNS → Email/Slack notificações para equipes de negócios
- Destino da API → Webhook para sistemas CRM/partner de gerenciamento externos

Exemplo

```
{
  "version": "1",
  "id": "<event id>",
  "detail-type": "Partner Connection Invitation Received",
  "source": "aws.partnercentral-account",
  "time": "<ISO 8601 date time>",
  "region": "us-east-1",
  "account": "<corresponding partner AWS account>",
  "resources": [ "<<Connection Invitation ARN>>" ],
```

```
"detail": {
  "catalog": "AWS",
  "connectionInvitation" :{
    "arn": "<<Connection Invitation ARN>>",
    "id": "pacinv-****",
    "connectionType": "OpportunityCollaboration",
    "inviterEmail": "abc@def.com",
    "invitationMessage": "We'd like to collaborate on a joint solution for cloud
security. Please accept this connection invitation to proceed.",
    "senderCompanyName": "<<Sender Partner Account Name>>",
    "senderProfileId": "pprofile-****",
    "expiresAt": "<ISO 8601 date time>"
  }
}
```

Convite de conexão de parceiro aceito

Contexto de acionamento

Esse evento é gerado quando um convite de conexão é aceito com sucesso e uma conexão é criada e mantida no armazenamento de dados do PAC por meio da `AcceptConnectionInvitation` API. O evento é enviado imediatamente após a conclusão bem-sucedida da aceitação do convite e o estabelecimento da conexão.

O evento será enviado automaticamente quando um convite de conexão for aceito. Aceito um evento por convite, sem eventos duplicados para a mesma aceitação.

Destinatários

Ambas as AWS contas envolvidas no convite de conexão: a conta do remetente que originalmente enviou o convite e a conta do destinatário que aceitou o convite.

Tratamento esperado

Ações típicas do cliente:

- Ativação de parceria: acione fluxos de trabalho para permitir atividades comerciais de camada 2
- Atualizações de status: atualize os sistemas de gerenciamento de parceiros com o status da conexão ativa
- Notificação: Alerta as equipes de negócios sobre o estabelecimento bem-sucedido da parceria

- Provisionamento de acesso: conceda automaticamente as permissões apropriadas para colaboração
- Analytics: acompanhe as taxas de conversão de parcerias e as métricas de sucesso

Padrões comuns de integração:

- Função Lambda → Ativar permissões de compartilhamento de dados e atualizar o portal do parceiro
- Fila SQS → Batch processa ativações de conexão para configuração de negócios
- Tópico do SNS → Email/Slack notificações para equipes técnicas e comerciais
- Destino da API → Webhook para sistemas de CRM para atualizar o status da parceria

Exemplo

```
{
  "version": "1",
  "id": "<event id>",
  "detail-type": "Partner Connection Invitation Accepted",
  "source": "aws.partnercentral-account",
  "time": "<ISO 8601 date time>",
  "region": "us-east-1",
  "resources": [ "<<Connection Invitation ARN>>" , "<<Connection ARN>>" ],
  "account": "<corresponding partner AWS account>",
  "detail": {
    "catalog": "AWS",
    "connectionInvitation" :{
      "arn": "<<Connection Invitation ARN>>",
      "id": "pacinv-****",
      "connectionType": "OpportunityCollaboration"
    },
    "connection": {
      "arn": "<<Connection ARN>>",
      "id": "pac-****",
      "Participant1AccountId": "<<Sender AWS AccountId>>",
      "Participant2AccountId": "<<Receiver AWS AccountId>>",
      "status": "ACTIVE"
    }
  }
}
```

Convite de conexão de parceiro rejeitado

Contexto de acionamento

Esse evento é gerado quando um convite de conexão é rejeitado com sucesso por meio da `RejectConnectionInvitation` API. O evento é enviado imediatamente após a rejeição do convite ser processada e persistida no armazenamento de dados do PAC.

O evento será enviado automaticamente quando um convite de conexão for rejeitado. Um evento por convite rejeitado sem eventos duplicados para a mesma rejeição.

Destinatários

A AWS conta que originalmente enviou o convite de conexão (a conta do remetente).

Tratamento esperado

Ações típicas do cliente:

- Atualizações de status: atualize os sistemas de gerenciamento de parceiros com o status de rejeição
- Follow-up Ações: acionar fluxos de trabalho para abordagens alternativas de parceria
- Notificação: alerte as equipes de negócios sobre a decisão de parceria
- Limpeza: remova referências de convites pendentes dos sistemas internos

Padrões comuns de integração:

- Função Lambda → Atualize o portal do parceiro e acione fluxos de trabalho de acompanhamento
- Fila SQS → Rejeições de processos em lote para análise de negócios
- Tópico do SNS → Email/Slack notificações para equipes de desenvolvimento de negócios
- Destino da API → Webhook para sistemas de CRM para atualizar o status da oportunidade

Exemplo

```
{
  "version": "1",
  "id": "<event id>",
  "detail-type": "Partner Connection Invitation Rejected",
  "source": "aws.partnercentral-account",
```

```
"time": "<ISO 8601 date time>",
"region": "us-east-1",
"account": "<corresponding partner AWS account>",
"resources": [ "<Connection Invitation ARN>" ],
"detail": {
  "catalog": "AWS",
  "connectionInvitation" :{
    "arn": "<Connection Invitation ARN>",
    "id": "pacinv-****",
    "connectionType": "OpportunityCollaboration",
    "invitationMessage": "We'd like to collaborate on a joint solution for cloud
security. Please accept this connection invitation to proceed.",
    "receiverProfileId": "pprofile-****"
  }
}
```

Conexão de parceiro cancelada

Contexto de acionamento

Esse evento é gerado quando uma conexão ativa é cancelada com sucesso e o status ativo é atualizado para encerrado no armazenamento de dados do PAC por meio da API. CancelConnection O evento é enviado imediatamente após o cancelamento da conexão ser processado e o status da conexão ser atualizado.

O evento será enviado automaticamente quando a conexão for encerrada. Um evento por conexão cancelado sem eventos duplicados para o mesmo cancelamento.

Destinatários

A AWS conta do outro participante na conexão (não daquele que iniciou o cancelamento).

Tratamento esperado

Ações típicas do cliente:

- Revogação de acesso: revogue imediatamente as permissões e desative o compartilhamento de dados
- Atualizações de status: atualize os sistemas de gerenciamento de parceiros com o status da conexão encerrada
- Notificação: Alerta as equipes técnicas e comerciais sobre o término da parceria

- Limpeza: remova as referências de conexão e limpe os recursos compartilhados
- Análise: acompanhe a duração da conexão e os padrões de encerramento

Padrões comuns de integração:

- Função Lambda → Revogar permissões e atualizar o status do portal do parceiro
- Fila SQS → Terminações de conexão de processamento em lote para fluxos de trabalho de limpeza
- Tópico do SNS → Email/Slack notificações para equipes técnicas e comerciais
- Destino da API → Webhook para sistemas de CRM para atualizar o status da parceria

Exemplo

```
{
  "version": "1",
  "id": "<event id>",
  "detail-type": "Partner Connection Cancelled",
  "source": "aws.partnercentral-account",
  "time": "<ISO 8601 date time>",
  "region": "us-east-1",
  "resources": [ "<<Connection ARN>>" ],
  "account": "<corresponding partner AWS account>",
  "detail": {
    "catalog": "AWS",
    "connection": {
      "arn": "<<Connection Invitation ARN>>",
      "id": "pac-****",
      "connectionType": "OpportunityCollaboration",
      "canceledBy": "<<AWS account ID>>"
    }
  }
}
```

Convite de conexão de parceiro cancelado

Contexto de acionamento

Esse evento é gerado quando um convite de conexão pendente é cancelado com sucesso por meio da `CancelConnectionInvitation` API. O evento é enviado imediatamente após o cancelamento do convite ser processado e o status do convite ser atualizado para CANCELADO.

O evento será enviado automaticamente quando um convite de conexão for cancelado. Um evento por convite de conexão foi cancelado sem eventos duplicados para o mesmo cancelamento.

Destinatários

A AWS conta que deveria receber o convite de conexão (a conta do destinatário).

Tratamento esperado

Ações típicas do cliente:

- Atualizações de status: atualize os sistemas de gerenciamento de parceiros para remover referências de convites pendentes
- Notificação: alerte as equipes de negócios de que uma possível oportunidade de parceria foi retirada
- Limpeza: remova as referências de convite dos sistemas internos de rastreamento
- Analytics: acompanhe os padrões de cancelamento de convites para obter informações sobre parcerias

Padrões comuns de integração:

- Função Lambda → Atualizar o portal do parceiro e remover notificações de convites pendentes
- Fila SQS → Cancelamentos de processos em lote para análise de negócios
- Tópico do SNS → Email/Slack notificações para equipes de desenvolvimento de negócios
- Destino da API → Webhook para sistemas de CRM para atualizar o status da oportunidade

Exemplo

```
{
  "version": "1",
  "id": "<event id>",
  "detail-type": "Partner Connection Invitation Cancelled",
  "source": "aws.partnercentral-account",
  "time": "<ISO 8601 date time>",
  "region": "us-east-1",
  "account": "<corresponding partner AWS account>",
  "resources": [ "<<Connection Invitation ARN>>" ],
  "detail": {
    "catalog": "AWS",
```

```
"connectionInvitation" :{
  "arn": "<<Connection Invitation ARN>>",
  "id": "pacinv-****",
  "connectionType": "OpportunityCollaboration",
  "invitationMessage": "We'd like to collaborate on a joint solution for cloud
security. Please accept this connection invitation to proceed.",
  "senderProfileId": "pprofile-****"
}
}
```

O convite de conexão de parceiro expirou

Contexto de acionamento

Esse evento é gerado quando um convite de conexão pendente expira automaticamente após atingir seu registro de ExpiresAt data e hora sem ser aceito ou rejeitado pelo destinatário. O evento é acionado automaticamente pelo sistema quando o prazo de expiração do convite é atingido.

O evento será enviado automaticamente quando o convite de conexão expirar. Um evento por convite expirou sem eventos duplicados com a mesma expiração.

Destinatários

Ambas as AWS contas envolvidas no convite de conexão: a conta do remetente que originalmente enviou o convite e a conta do destinatário que recebeu o convite.

Tratamento esperado

Ações típicas do cliente:

- Atualizações de status: atualize os sistemas de gerenciamento de parceiros para marcar o convite como expirado
- Follow-up Ações: acionar fluxos de trabalho para reengajamento ou abordagens alternativas de parceria
- Notificação: alerte as equipes de negócios sobre oportunidades de parceria perdidas
- Limpeza: remova referências de convite expiradas dos sistemas internos de rastreamento

Padrões comuns de integração:

- Função Lambda → Atualize o portal do parceiro e acione fluxos de trabalho de acompanhamento

- Fila SQS → Expirações de processos em lote para análise de negócios
- Tópico do SNS → Email/Slack notificações para equipes de desenvolvimento de negócios
- Destino da API → Webhook para sistemas de CRM para atualizar o status da oportunidade

Exemplo

```
{
  "version": "1",
  "id": "<event id>",
  "detail-type": "Partner Connection Invitation Expired",
  "source": "aws.partnercentral-account",
  "time": "<ISO 8601 date time>",
  "region": "us-east-1",
  "resources": [ "<<Connection Invitation ARN>>" ],
  "detail": {
    "catalog": "AWS",
    "connectionInvitation": {
      "arn": "<<Connection Invitation ARN>>",
      "id": "pacinv-****",
      "connectionType": "OpportunityCollaboration",
      "inviterEmail": "abc@def.com",
      "invitationMessage": "We'd like to collaborate on a joint solution for cloud security. Please accept this connection invitation to proceed.",
      "senderProfileId": "pprofile-****",
      "receiverProfileId": "pprofile-****"
    }
  }
}
```

Notificações para o AWS API de vendas da Central de Parceiros

Os eventos da API de vendas fornecem notificações em tempo real sobre mudanças no status ou nos detalhes das oportunidades. Esses eventos ajudam a manter seus sistemas sincronizados com o AWS Partner Central e ajudam a garantir respostas e atualizações oportunas.

Tópicos

- [Preencha o pré-requisito para monitorar eventos](#)
- [Configure EventBridge a Amazon para monitorar eventos](#)
- [Saiba mais sobre a venda de eventos de API](#)

Preencha o pré-requisito para monitorar eventos

Os usuários precisam das permissões apropriadas do IAM para acessar e gerenciar eventos publicados pela API de vendas do Partner Central. Para obter mais informações sobre as ações, os recursos e as chaves de condição disponíveis para EventBridge, consulte [Usando as condições da política do IAM na Amazon EventBridge](#) no Guia EventBridge do usuário da Amazon. Uma das chaves de condição é `events:detail-type`, que pode ser usada para definir permissões para tipos específicos de eventos.

O exemplo de política a seguir demonstra como personalizar e definir o escopo das permissões para os eventos propostos. A `AllowPutRuleForPartnercentralSellingEvents` declaração permite a criação de regras, mas somente para eventos da `aws.partnercentral-selling` fonte.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowPutRuleForPartnercentralSellingEvents",
      "Effect": "Allow",
      "Action": "events:PutRule",
      "Resource": "*",
      "Condition": {
        "ForAllValues:StringEquals": {
          "events:source": "aws.partnercentral-selling"
        }
      }
    }
  ]
}
```

Configure EventBridge a Amazon para monitorar eventos

Para monitorar a venda de eventos de API, você cria uma EventBridge regra que corresponde aos eventos que você deseja capturar. Você pode usar o Console de gerenciamento da AWS ou os AWS SDKs para criar e gerenciar regras. As seções a seguir explicam como criar regras usando os dois métodos. Independentemente do método usado, você deve criar a regra na `us-east-1` região Leste dos EUA (Norte da Virgínia).

Console de gerenciamento da AWS configuração

Para configurar uma EventBridge regra usando o Console de gerenciamento da AWS, siga as etapas em [Criação de regras que reagem a eventos na Amazon EventBridge](#) no Guia EventBridge do usuário da Amazon. Ao criar a regra, você deve definir o barramento de eventos como padrão e criar a regra na us-east-1 região Leste dos EUA (Norte da Virgínia).

Veja a seguir um exemplo de uma regra de evento:

```
{
  "source": ["aws.partnercentral-selling"],
  "detail": {
    "catalog": ["AWS"]
  }
}
```

AWS Configuração do SDK

Você pode usar os AWS SDKs para criar e gerenciar EventBridge regras de forma programática. Para obter mais informações, consulte [PutRule](#) a Amazon EventBridge API Reference.

O exemplo a seguir usa o AWS SDK para Python (Boto3):

```
import boto3

client = boto3.client('events', region_name='us-east-1')

response = client.put_rule(
    Name='MyOpportunityCreatedRule',
    EventPattern=
    '{
      "source": ["aws.partnercentral-selling"],
      "detail-type": ["Opportunity Created"],
      "detail": {"catalog": ["AWS"]}
    }',
    State='ENABLED'
)
print('Rule ARN:', response['RuleArn'])
```

Saiba mais sobre a venda de eventos de API

As seções a seguir descrevem os tipos de eventos da API de venda, os cenários que os acionam e os exemplos de eventos.

Tipos de eventos

A seguir estão os tipos de eventos e seus gatilhos.

- [Oportunidade criada](#): acionada quando uma nova oportunidade é criada.
- [Oportunidade atualizada](#): acionada quando uma oportunidade (oportunidade ou seu resumo de AWS oportunidade correspondente) é atualizada.
- [Convite de engajamento criado](#): acionado quando um convite de AWS referência é criado.
- [Convite de engajamento aceito](#): acionado quando um parceiro aceita um convite de AWS engajamento, confirmando seu interesse em colaborar com a AWS oportunidade.
- [Convite de noivado rejeitado](#): acionado quando um parceiro rejeita um convite de AWS noivado.
- O [convite de engajamento expirou](#): Este evento é acionado quando um parceiro rejeita um. EngagementInvitation Ele notifica o parceiro de envio e recebimento de que o convite expirou.
- [Membro de engajamento adicionado](#): esse evento é acionado quando um novo membro se junta ao Engajamento após aceitar um convite. Ele notifica todos os membros atuais do Engajamento sobre o novo membro que está sendo adicionado ao Engajamento.
- [Criação de instantâneo do recurso de engajamento](#): esse evento é acionado quando a nova revisão do instantâneo dos recursos (como oportunidades) associados a um compromisso é criada. Ele notifica todos os membros atuais do Engajamento sobre as alterações no recurso Snapshot associado.
- [Engajamento criado](#): acionado quando um novo engajamento é criado.
- [Engajamento atualizado](#): acionado quando um engajamento é atualizado.

Cenários de eventos

A tabela a seguir descreve como os eventos operam em diferentes cenários. As seguintes suposições são feitas para esses cenários:

- AWS também é parceira nesses cenários.
- ResourceSnapshotJob está configurado corretamente pelo parceiro.

- Os campos atualizados sobre a oportunidade também estão no modelo de oportunidade do Resource Snapshot.

Ação de você como parceiro	Eventos recebidos	Tipo de participante
Você cria uma oportunidade	Oportunidade criada	
Você usa <code>StartEngagementFromOpportunityTask</code> para enviar uma oportunidade	Oportunidade atualizada, engajamento criado, resumo do recurso de engajamento criado, membro do engajamento adicionado, convite de engajamento criado	
AWS aprova sua oportunidade enviada	Convite de engajamento aceito, membro de engajamento adicionado, oportunidade atualizada, resumo do recurso de engajamento criado	
AWS rejeita sua oportunidade enviada	Convite de engajamento rejeitado, oportunidade atualizada, resumo do recurso de engajamento criado	
Você associa uma AWS Marketplace oferta a uma oportunidade	Oportunidade atualizada	
Você associa uma solução a uma oportunidade	Oportunidade atualizada, resumo dos recursos de engajamento criado	
Você atualiza uma oportunidade	Oportunidade atualizada, instantâneo do recurso de engajamento criado (opcional se <code>ResourceSnapshotJob</code> estiver configurado e	

Ação de você como parceiro	Eventos recebidos	Tipo de participante
	as atualizações de campo estiverem no modelo)	
AWS atualiza sua oportunidade	Oportunidade atualizada, resumo dos recursos de engajamento criado	
Você convida outro parceiro para um compromisso	Convite de engajamento criado	Remetente
Seu convite para participar de um compromisso é aceito por um parceiro	Convite de noivado aceito, membro de engajamento adicionado	Remetente
Seu convite para participar de um compromisso é rejeitado por um parceiro	Convite de compromisso rejeitado	Remetente
Seu convite para participar de um compromisso não é atendido por um parceiro por 15 dias	O convite de noivado expirou	Remetente
Você recebe um convite de outro parceiro para participar de um compromisso	Convite de engajamento criado	Receptor
Você aceita o convite de outro parceiro para participar de um compromisso via <code>StartEngagementByAcceptingInvitationTask</code>	Convite de engajamento aceito, membro de engajamento adicionado, oportunidade criada, oportunidade atualizada, instantâneo do recurso de engajamento criado	Receptor
Você rejeita o convite de outro parceiro para participar de um compromisso	Convite de compromisso rejeitado	Receptor

Ação de você como parceiro	Eventos recebidos	Tipo de participante
Você não age sob convite para participar de um compromisso de outro parceiro por 15 dias	O convite de noivado expirou	Receptor
Você cria um compromisso	Engajamento criado	Remetente
Você atualiza um compromisso	Engajamento atualizado	Remetente, destinatário

Eventos de exemplo

As seções a seguir fornecem exemplos dos eventos listados anteriormente na seção anterior.

Tópicos

- [Oportunidade criada](#)
- [Oportunidade atualizada](#)
- [Convite de engajamento criado](#)
- [Convite de compromisso aceito](#)
- [Convite de compromisso rejeitado](#)
- [O convite de noivado expirou](#)
- [Membro de engajamento adicionado](#)
- [Instantâneo do recurso de engajamento criado](#)
- [Engajamento criado](#)
- [Engajamento atualizado](#)

Oportunidade criada

Os parceiros usam esse evento para:

- Notifique seus usuários de que uma nova oportunidade foi criada.
- Acione fluxos de trabalho automatizados, como atualizar os sistemas de CRM ou notificar as equipes de vendas sobre a criação de novas oportunidades.

O exemplo a seguir mostra um evento típico de Oportunidade Criada.

```
{
  "version": "1",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "source": "aws.partnercentral-selling",
  "detail-type": "Opportunity Created",
  "time": "2023-04-16T15:23:45Z",
  "region": "us-east-1",
  "account": "123456789012",
  "detail": {
    "schemaVersion": "<version number>",
    "catalog": "<Sandbox | AWS>",
    "opportunity": {
      "identifier": "String"
    }
  }
}
```

Oportunidade atualizada

Os parceiros usam esse evento para:

- Notifique seus usuários de que uma oportunidade existente foi atualizada.
- Acione fluxos de trabalho automatizados, como atualizar sistemas de CRM ou notificar equipes de vendas.

O exemplo a seguir mostra um evento típico de atualização de oportunidade.

```
{
  "version": "1",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "source": "aws.partnercentral-selling",
  "detail-type": "Opportunity Updated",
  "time": "2023-04-16T15:23:45Z",
  "region": "us-east-1",
  "account": "123456789012",
  "detail": {
    "schemaVersion": "<version number>",
    "catalog": "<Sandbox | AWS>",
    "opportunity": {
```

```

        "identifier": "String"
    }
}

```

Convite de engajamento criado

Os parceiros usam esse evento para:

- Notifique seus usuários sobre as novidades que EngagementInvitation eles receberam.
- Acione fluxos de trabalho automatizados para aceitar ou rejeitar o convite, como atualizar sistemas de CRM ou notificar equipes de vendas.

O exemplo a seguir mostra um evento típico criado por convite de engajamento.

```

{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "Engagement Invitation Created",
  "source": "aws.partnercentral-selling",
  "account": "123456789012",
  "time": "2023-04-15T12:34:56Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:partnercentral:us-east-1::catalog/AWS/engagement-invitation/engi-
v7p8z56whnauo"
  ],
  "detail": {
    "catalog": "AWS",
    "engagementInvitation": {
      "arn": "arn:aws:partnercentral:us-east-1::catalog/AWS/engagement-invitation/
engi-v7p8z56whnauo",
      "id": "engi-v7p8z56whnauo",
      "engagementId": "eng-12345678901234",
      "senderAccountId": "string",
      "receiverAccountId": "string",
      "senderCompanyName": "string",
      "expirationDate": "string",
      "participantType": "Enum", //Sender/Receiver
      "payloadType": "LeadInvitation" //OpportunityInvitation|LeadInvitation
    }
  }
}

```

```
}
```

Convite de compromisso aceito

Os parceiros usam esse evento para:

- Notifique seus usuários de que o EngagementInvitation foi aceito.
- Atualize seus registros internos para refletir o novo parceiro no Engajamento.
- Acione fluxos de trabalho automatizados para sincronizar dados ou notificar outros membros da equipe sobre o novo membro do Engagement.

O exemplo a seguir mostra um evento típico aceito por convite de noivado.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "Engagement Invitation Accepted",
  "source": "aws.partnercentral-selling",
  "account": "123456789012",
  "time": "2023-04-16T15:23:45Z",
  "region": "us-east-1",
  "resources": ["arn:aws:partnercentral:us-east-1::catalog/AWS/engagement-invitation/engi-v7p8z56whnauo"],
  "detail": {
    "catalog": "AWS",
    "engagementInvitation": {
      "arn": "arn:aws:partnercentral:us-east-1::catalog/AWS/engagement-invitation/engi-v7p8z56whnauo",
      "id": "engi-v7p8z56whnauo",
      "engagementId": "eng-12356whnauo",
      "participantType": "Enum", //Sender/Receiver
      "senderAccountId": "string",
      "receiverAccountId": "string",
      "payloadType": "LeadInvitation" //OpportunityInvitation|LeadInvitation
    }
  }
}
```

Convite de compromisso rejeitado

Os parceiros usam esse evento para:

- Notifique seus usuários de que o EngagementInvitation foi rejeitado.
- Atualize seus registros internos para refletir o convite rejeitado.
- Acione todos os fluxos de trabalho necessários, como notificar a equipe de vendas sobre a rejeição.

O exemplo a seguir mostra um evento típico de rejeição de convite de engajamento.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "Engagement Invitation Rejected",
  "source": "aws.partnercentral-selling",
  "account": "123456789012",
  "time": "2023-04-17T09:48:27Z",
  "region": "us-east-1",
  "resources": ["arn:aws:partnercentral:us-east-1::catalog/AWS/engagement-invitation/engi-v7p8z56whnauo"],
  "detail": {
    "catalog": "AWS",
    "engagementInvitation": {
      "arn": "arn:aws:partnercentral:us-east-1::catalog/AWS/engagement-invitation/engi-v7p8z56whnauo",
      "id": "engi-v7p8z56whnauo",
      "engagementId": "eng-12356whnauo",
      "participantType": "Enum", //Sender/Receiver
      "senderAccountId": "string",
      "receiverAccountId": "string",
      "payloadType": "LeadInvitation" //OpportunityInvitation|LeadInvitation
    }
  }
}
```

O convite de noivado expirou

Os parceiros usam esse evento para:

- Notifique seus usuários de que o EngagementInvitation expirou e não pode mais ser usado.
- Atualize seus registros internos para refletir o convite expirado.
- Acione todos os fluxos de trabalho necessários, como notificar a equipe de vendas sobre o convite expirado.

O exemplo a seguir mostra um evento típico de convite de engajamento expirado.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "Engagement Invitation Expired",
  "source": "aws.partnercentral-selling",
  "account": "123456789012",
  "time": "2023-04-18T18:20:15Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:partnercentral:us-east-1::catalog/AWS/engagement-invitation/engi-
v7p8z56whnauo"
  ],
  "detail": {
    "catalog": "AWS",
    "engagementInvitation": {
      "arn": "arn:aws:partnercentral:us-east-1::catalog/AWS/engagement-invitation/
engi-v7p8z56whnauo",
      "id": "engi-v7p8z56whnauo",
      "engagementId": "eng-12356whnauo",
      "participantType": "Enum", //Sender/Receiver
      "senderAccountId": "string",
      "receiverAccountId": "string",
      "payloadType": "LeadInvitation" //OpportunityInvitation|LeadInvitation
    }
  }
}
```

Membro de engajamento adicionado

Os parceiros usam esse evento para:

- Notifique seus usuários sobre as mudanças na associação do Engagement.
- Atualize seus registros internos para refletir o novo membro do Engagement.
- Acione todos os fluxos de trabalho necessários, como notificar os membros da equipe sobre as mudanças.

O exemplo a seguir mostra um evento típico de Engagement Member Added.

```
{
```

```

"version": "0",
"id": "01234567-0123-0123-0123-0123456789ab",
"detail-type": "Engagement Member Added",
"source": "aws.partnercentral-selling",
"account": "123456789012",
"time": "2023-04-16T15:23:45Z",
"region": "us-east-1",
"resources": [
  "arn:aws:partnercentral:us-east-1::catalog/AWS/engagement/eng-v7p8z56whnauo"
],
"detail": {
  "catalog" : "AWS",
  "engagement": {
    "id": "eng-v7p8z56whnauo"
  },
  "engagementMember": {
    "accountId": "string",
    "companyName": "string"
  }
}
}

```

Instantâneo do recurso de engajamento criado

Os parceiros usam esse evento para monitorar alterações nos recursos associados a um compromisso e acionar fluxos de trabalho automatizados, como atualizar sistemas de CRM ou notificar equipes de vendas. O modelo de snapshot determina o tipo de atualização:

OpportunitySummaryView

- Notifique todos os membros de um compromisso de que o compromisso foi atualizado.

AwsOpportunitySummaryFullView

- Acompanhe as atualizações feitas especificamente pela AWS para oportunidades dentro de um compromisso.
- Essa notificação e esse instantâneo são visíveis somente para o proprietário da oportunidade.
- Inclui atualizações de dimensionamento de negócios, que não estão disponíveis por meio do OpportunitySummaryView modelo.

O exemplo a seguir mostra um evento típico do Engagement Resource Snapshot Created.

```
{
```

```

    "version": "0",
    "id": "01234567-0123-0123-0123-0123456789ab",
    "detail-type": "Engagement Resource Snapshot Created",
    "source": "aws.partnercentral-selling",
    "account": "123456789012",
    "time": "2023-04-18T18:20:15Z",
    "region": "us-east-1",
    "resources": [
      "arn:aws:partnercentral:us-east-1::catalog/AWS/engagement/eng-v7p8z56whnauo"
    ],
    "detail": {
      "catalog" : "AWS",
      "resourceSnapshot": {
        "arn": "arn:aws:partnercentral-selling:us-east-1::catalog/AWS/engagement/eng-v7p8z56whnauo/resource/Opportunity/o-12312/template/temp-name/snapshot/snapshot-19232",
        "engagementId": "eng-v7p8z56whnauo",
        "resourceType": "Opportunity",
        "resourceId" : "0123211231",
        "createdBy": "123123123123",
        "targetMemberAccounts": ["123123123123", "aws"],
        "resourceSnapshotTemplateName": "temp-name"
      }
    }
  }
}

```

Engajamento criado

Os parceiros usam esse evento para:

- Notifique seus usuários de que um novo Engajamento foi criado.
- Acione fluxos de trabalho automatizados, como atualizar os sistemas de CRM ou notificar as equipes de vendas sobre novas criações de engajamento.

O exemplo a seguir mostra um evento típico de Engagement Created.

```

{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "Engagement Created",
  "source": "aws.partnercentral-selling",
  "account": "123456789012",
  "time": "2023-04-18T18:20:15Z",

```

```

    "region": "us-east-1",
    "resources": [
      "arn:aws:partnercentral:us-east-1::catalog/AWS/engagement/eng-567a1j5hxp351o"
    ],
    "detail": {
      "catalog": "AWS",
      "engagement": {
        "engagementArn": "arn:aws:partnercentral:us-east-1::catalog/AWS/engagement/eng-567a1j5hxp351o",
        "engagementId": "eng-567a1j5hxp351o",
        "CreatedAt": "2023-04-18T18:20:15Z",
        "CreatedBy": "aws",
        "ContextTypes": ["Lead"]
      }
    }
  }
}

```

Engajamento atualizado

Os parceiros usam esse evento para:

- Notifique seus usuários de que um Engajamento existente foi atualizado.
- Acione fluxos de trabalho automatizados, como atualizar sistemas de CRM ou notificar equipes de vendas.

O exemplo a seguir mostra um evento típico de Engagement Updated.

```

{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "Engagement Updated",
  "source": "aws.partnercentral-selling",
  "account": "123456789012",
  "time": "2023-04-18T18:20:15Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:partnercentral:us-east-1::catalog/AWS/engagement/eng-567a1j5hxp351o"
  ],
  "detail": {
    "catalog": "AWS",
    "engagement": {

```

```
    "engagementArn": "arn:aws:partnercentral:us-east-1::catalog/AWS/engagement/eng-567a1j5hxp351o",
    "engagementId": "eng-567a1j5hxp351o",
    "LastModifiedAt": "2023-04-18T18:20:15Z",
    "LastModifiedBy": "aws",
    "ContextTypes": ["Lead", "CustomerProject"]
  }
}
```

Notificações para o AWS API de benefícios do Partner Central

O Serviço de Benefícios oferece EventBridge eventos para BenefitApplications BenefitAllocations e. Esses eventos permitem que os clientes criem arquiteturas abrangentes orientadas por eventos que respondem às mudanças em todo o ciclo de vida dos recursos, desde a aplicação inicial até a alocação e o atendimento final.

Tópicos

- [Preencha o pré-requisito para monitorar eventos](#)
- [Configure EventBridge a Amazon para monitorar eventos](#)
- [Saiba mais sobre os eventos da API de benefícios](#)

Preencha o pré-requisito para monitorar eventos

Os usuários precisam das permissões apropriadas do IAM para acessar e gerenciar eventos publicados pela API de benefícios. Para obter mais informações sobre as ações, os recursos e as chaves de condição disponíveis para EventBridge, consulte [Usando as condições da política do IAM na Amazon EventBridge](#) no Guia EventBridge do usuário da Amazon. Uma das chaves de condição é `events:detail-type`, que pode ser usada para definir permissões para tipos específicos de eventos.

O exemplo de política a seguir demonstra como personalizar e definir o escopo das permissões para os eventos propostos. A `AllowPutRuleForPartnercentralBenefitsEvents` declaração permite a criação de regras, mas somente para eventos da `aws.partnercentral-benefits` fonte.

Para exemplos detalhados de políticas do IAM, consulte a documentação da AWS sobre EventBridge permissões.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowPutRuleForPartnerCentralBenefitsEvents",
      "Effect": "Allow",
      "Action": "events:PutRule",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "events:source": "aws.partnercentral-benefits"
        }
      }
    }
  ]
}
```

Configure EventBridge a Amazon para monitorar eventos

Para monitorar eventos da API de benefícios, você cria uma EventBridge regra que corresponde aos eventos que você deseja capturar. Você pode usar o Console de gerenciamento da AWS ou os AWS SDKs para criar e gerenciar regras. As seções a seguir explicam como criar regras usando os dois métodos. Independentemente do método usado, você deve criar a regra na us-east-1 região Leste dos EUA (Norte da Virgínia).

Console de gerenciamento da AWS configuração

Para configurar uma EventBridge regra usando o Console de gerenciamento da AWS, siga as etapas em [Criação de regras que reagem a eventos na Amazon EventBridge](#) no Guia EventBridge do usuário da Amazon. Ao criar a regra, você deve definir o barramento de eventos como padrão e criar a regra na us-east-1 região Leste dos EUA (Norte da Virgínia).

Veja a seguir um exemplo de uma regra de evento:

```
{
  "source": ["aws.partnercentral-benefits"],
  "detail": {
    "catalog": ["AWS"]
  }
}
```

AWS Configuração do SDK

Você pode usar os AWS SDKs para criar e gerenciar EventBridge regras de forma programática. Para obter mais informações, consulte [PutRule](#) a Amazon EventBridge API Reference.

O exemplo a seguir usa o AWS SDK para Python (Boto3):

```
import boto3

client = boto3.client('events', region_name='us-east-1')

response = client.put_rule(
    Name='MyBenefitApplicationCreatedRule',
    EventPattern=
    '{
        "source": ["aws.partnercentral-benefits"],
        "detail-type": ["Benefit Application Created"],
        "detail": {"catalog": ["AWS"]}
    }',
    State='ENABLED'
)
print('Rule ARN:', response['RuleArn'])
```

Saiba mais sobre os eventos da API de benefícios

As seções a seguir descrevem os tipos de eventos da API de benefícios, os cenários que os acionam e os exemplos de eventos.

Tipos de eventos

A seguir estão os tipos de eventos e seus gatilhos.

Aplicações de benefícios

- [Aplicativo de benefícios criado](#): quando um aplicativo de benefícios é criado
- [Solicitação de benefícios em análise](#): Quando uma solicitação de benefício é enviada ou quando a solicitação passa da Ação necessária para A análise, esse evento é acionado.
- [Ação de solicitação de benefícios necessária](#): Esse evento ocorre quando um revisor interno atualiza o status da inscrição para indicar que o parceiro precisa fornecer informações adicionais, esclarecimentos ou modificações antes que a análise possa continuar.

- [Solicitação de benefícios rejeitada](#): esse evento ocorre quando um revisor interno atualiza o status da solicitação para indicar que a solicitação não atende aos critérios ou requisitos do benefício e foi recusada
- [Solicitação de benefícios aprovada](#): Esse evento ocorre quando um revisor interno atualiza o status da solicitação para indicar que a solicitação foi aprovada para alocação de benefícios.
- [Estágio de solicitação de benefícios alterado](#): Esse evento ocorre quando um revisor interno atualiza o estágio da solicitação durante processos de revisão, como análise comercial, AWS revisão, análise financeira etc.

Alocações de benefícios

- [Alocação de benefícios ativada](#): quando uma alocação de benefícios é criada ou atualizada de volta para Ativada por um serviço interno.
- [Alocação de benefícios inativada](#): esse evento ocorre quando um revisor interno atualiza o status da alocação ou quando uma alocação de benefícios expira automaticamente.
- [Alocação de benefícios cumprida](#): esse evento ocorre quando um revisor interno atualiza o status da alocação para indicar que a alocação foi totalmente utilizada.

Eventos de exemplo

As seções a seguir fornecem exemplos dos eventos listados anteriormente na seção anterior.

Tópicos

- [Aplicativo de benefícios criado](#)
- [Solicitação de benefícios em análise](#)
- [Ação de solicitação de benefícios necessária](#)
- [Pedido de benefício rejeitado](#)
- [Solicitação de benefícios aprovada](#)
- [Estágio de solicitação de benefícios alterado](#)
- [Alocação de benefícios ativada](#)
- [Alocação de benefícios inativada](#)
- [Alocação de benefícios cumprida](#)

Aplicativo de benefícios criado

Contexto de acionamento

O evento Benefit Application Created é publicado na Amazon EventBridge quando um novo Pedido de Benefícios é criado com sucesso por meio do Serviço de Benefícios. Esse evento ocorre quando a chamada da CreateBenefitApplication API é concluída com êxito, indicando que um parceiro iniciou uma solicitação para um benefício específico.

Esse evento representa o início do ciclo de vida da solicitação de benefícios e fornece aos clientes uma notificação imediata quando novas solicitações são enviadas para suas contas.

Destinatários

A conta proprietária receberá o evento para a alocação de benefícios.

Exemplo

```
{
  "id": "7gh88765-k0jh-d08g-i292-2ff1796m030n",
  "detail-type": "Benefit Application Created",
  "source": "aws.partnercentral-benefits",
  "account": "123456789012",
  "time": "2025-02-10T15:45:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:partnercentral:us-east-1:123456789012:benefit-application/benappl-12345abcdef123"
  ],
  "detail": {
    "catalog": "AWS",
    "benefitApplication": {
      "id": "benappl-12345abcdef123",
      "fulfillmentTypes": ["CREDITS"],
      "status": "PENDING_SUBMISSION",
      "revision": "hh7e8400-e29b-41d4-a716-446655440012"
    },
    "benefit": {
      "id": "ben-xyz789uvw012",
      "name": "AWS Partner Credits",
      "program": ["AWS Partner Network"]
    }
  }
}
```

```
}
```

Solicitação de benefícios em análise

Contexto de acionamento

O evento Benefit Application In Review é publicado na Amazon EventBridge quando um aplicativo de benefícios passa para o status IN_REVIEW por meio do Benefits Service. Esse evento ocorre quando:

- Um parceiro envia sua inscrição por meio da SubmitBenefitApplication chamada de API
- Um aplicativo faz a transição de ACTION_REQUIRED de volta para IN_REVIEW depois que o parceiro responde ao feedback
- Uma solicitação é recuperada e, em seguida, reenviada.

Esse evento indica que a inscrição entrou no processo formal de revisão e está sendo avaliada por equipes AWS internas.

Destinatários

A conta proprietária receberá o evento para a alocação de benefícios.

Exemplo

```
{
  "id": "8hi99876-l1ki-e19h-j303-3gg2807n141o",
  "detail-type": "Benefit Application In Review",
  "source": "aws.partnercentral-benefits",
  "account": "123456789012",
  "time": "2025-02-10T16:00:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:partnercentral:us-east-1:123456789012:benefit-application/benappl-12345abcdef123"
  ],
  "detail": {
    "catalog": "AWS",
    "benefitApplication": {
      "id": "benappl-12345abcdef123",
      "fulfillmentTypes": ["CREDITS"],
      "status": "IN_REVIEW",
```

```

    "stage": "BUSINESS_REVIEW",
    "revision": "ii8f9511-f30c-52e5-b827-557766551123"
  },
  "benefit": {
    "id": "ben-xyz789uvw012",
    "name": "AWS Partner Credits",
    "program": ["AWS Partner Network"]
  }
}
}
}

```

Ação de solicitação de benefícios necessária

Contexto de acionamento

O evento Benefit Application Action Required é publicado na Amazon EventBridge quando um aplicativo de benefícios faz a transição para o status ACTION_REQUIRED por meio do Benefits Service. Esse evento ocorre quando um revisor interno atualiza o status do aplicativo por meio da UpdateBenefitApplicationInternal API para indicar que o parceiro precisa fornecer informações adicionais, esclarecimentos ou modificações antes que a análise possa continuar.

Esse evento representa um ponto crítico no ciclo de vida do aplicativo em que a intervenção do parceiro é necessária para levar o aplicativo adiante no processo de análise.

Destinatários

A conta proprietária receberá o evento para a alocação de benefícios

Exemplo

```

{
  "id": "9ij00987-m2lj-f20i-k414-4hh3918o252p",
  "detail-type": "Benefit Application Action Required",
  "source": "aws.partnercentral-benefits",
  "account": "123456789012",
  "time": "2025-02-12T10:30:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:partnercentral:us-east-1:123456789012:benefit-application/benappl-12345abcdef123"
  ],
  "detail": {

```

```
"catalog": "AWS",
"benefitApplication": {
  "id": "benappl-12345abcdef123",
  "fulfillmentTypes": ["CREDITS"],
  "status": "ACTION_REQUIRED",
  "stage": "BUSINESS_REVIEW",
  "revision": "jj9g0622-g41d-63f6-c938-668877662234"
},
"benefit": {
  "id": "ben-xyz789uvw012",
  "name": "AWS Partner Credits",
  "program": ["AWS Partner Network"]
}
}
```

Pedido de benefício rejeitado

Contexto de acionamento

O evento Solicitação de Benefício Rejeitada é publicado na Amazon EventBridge quando uma Solicitação de Benefício passa para o status REJEITADA por meio do Serviço de Benefícios. Esse evento ocorre quando um revisor interno atualiza o status do aplicativo por meio da UpdateBenefitApplicationInternal API para indicar que o aplicativo não atende aos critérios ou requisitos de benefícios e foi recusado.

Esse evento representa um estado terminal no ciclo de vida do aplicativo em que o aplicativo foi formalmente recusado e não prosseguirá com a alocação de benefícios.

Destinatários

A conta proprietária receberá o evento para a alocação de benefícios.

Exemplo

```
{
  "id": "0kl11098-n3mk-g31j-1525-5ii4029p363q",
  "detail-type": "Benefit Application Rejected",
  "source": "aws.partnercentral-benefits",
  "account": "123456789012",
  "time": "2025-02-15T14:20:00Z",
  "region": "us-east-1",
  "resources": [
```

```
"arn:aws:partnercentral:us-east-1:123456789012:benefit-application/
benappl-12345abcdef123"
],
"detail": {
  "catalog": "AWS",
  "benefitApplication": {
    "id": "benappl-12345abcdef123",
    "fulfillmentTypes": ["CREDITS"],
    "status": "REJECTED",
    "revision": "kk0h1733-h52e-74g7-d049-779988773345"
  },
  "benefit": {
    "id": "ben-xyz789uvw012",
    "name": "AWS Partner Credits",
    "program": ["AWS Partner Network"]
  }
}
}
```

Solicitação de benefícios aprovada

Contexto de acionamento

O evento Solicitação de Benefício Aprovada é publicado na Amazon EventBridge quando uma Solicitação de Benefício passa para o status APROVADO por meio do Serviço de Benefícios. Esse evento ocorre quando um revisor interno atualiza o status do aplicativo por meio da UpdateBenefitApplicationInternal API para indicar que o aplicativo atende a todos os critérios e requisitos de benefícios e foi aprovado para alocação de benefícios.

Esse evento representa uma conclusão bem-sucedida do processo de análise da inscrição e normalmente aciona a criação de uma alocação de benefícios correspondente.

Destinatários

A conta proprietária receberá o evento para a alocação de benefícios.

Exemplo

```
{
  "id": "1lm22109-o4nl-h42k-m636-6jj5130q474r",
  "detail-type": "Benefit Application Approved",
  "source": "aws.partnercentral-benefits",
  "account": "123456789012",
```

```

"time": "2025-02-14T16:45:00Z",
"region": "us-east-1",
"resources": [
  "arn:aws:partnercentral:us-east-1:123456789012:benefit-application/
benappl-12345abcdef123"
],
"detail": {
  "catalog": "AWS",
  "benefitApplication": {
    "id": "benappl-12345abcdef123",
    "fulfillmentTypes": ["CREDITS"],
    "status": "APPROVED",
    "revision": "l11i2844-i63f-85h8-e150-880099884456"
  },
  "benefit": {
    "id": "ben-xyz789uvw012",
    "name": "AWS Partner Credits",
    "program": ["AWS Partner Network"]
  }
}
}

```

Estágio de solicitação de benefícios alterado

Contexto de acionamento

O evento Benefit Application Stage Changed é publicado na Amazon EventBridge quando o estágio de análise de um pedido de benefícios é atualizado por meio do Serviço de Benefícios. Esse evento ocorre quando um revisor interno atualiza o estágio do aplicativo para um dos possíveis valores de Enum (FINANCE_REVIEW, BUSINESS_REVIEW, TECH_REVIEW e _REVIEW).AWS

Diferentemente das mudanças de status, as mudanças de estágio são atualizações informativas somente para leitura que fornecem visibilidade da progressão do fluxo de trabalho de revisão interna sem exigir a ação do parceiro.

Destinatários

A conta proprietária receberá o evento para a alocação de benefícios.

Exemplo

```

{
  "id": "2mn33210-p5om-i53l-n747-7kk6241r585s",

```

```

"detail-type": "Benefit Application Stage Changed",
"source": "aws.partnercentral-benefits",
"account": "123456789012",
"time": "2025-02-13T11:15:00Z",
"region": "us-east-1",
"resources": [
  "arn:aws:partnercentral:us-east-1:123456789012:benefit-application/
benappl-12345abcdef123"
],
"detail": {
  "catalog": "AWS",
  "benefitApplication": {
    "id": "benappl-12345abcdef123",
    "fulfillmentTypes": ["CREDITS"],
    "status": "IN_REVIEW",
    "stage": "FINANCIAL_REVIEW",
    "revision": "mm2j3955-j74g-96i9-f261-991100995567"
  },
  "benefit": {
    "id": "ben-xyz789uvw012",
    "name": "AWS Partner Credits",
    "program": ["AWS Partner Network"]
  }
}
}

```

Alocação de benefícios ativada

Contexto de acionamento

O evento Benefit Allocation Activated é publicado na Amazon EventBridge quando uma alocação de benefícios passa para o status ATIVO por meio do Benefits Service. Esse evento ocorre quando um revisor interno atualiza o status da alocação por meio da UpdateBenefitAllocationInternal API para reativar uma alocação anteriormente inativa.

Esse evento geralmente ocorre quando uma alocação que foi temporariamente inativada (devido à expiração, mudanças na política ou motivos administrativos) é restaurada ao status de ativa, disponibilizando novamente os benefícios para utilização pelo parceiro.

Destinatários

A AWS conta pertencente ao parceiro receberá a mensagem. Em outras palavras, a AWS Conta que possui o recurso.

Evento de exemplo

```
{
  "id": "3no44321-q6pn-j64m-o858-8117352s696t",
  "detail-type": "Benefit Allocation Activated",
  "source": "aws.partnercentral-benefits",
  "account": "123456789012",
  "time": "2025-03-01T08:15:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:partnercentral:us-east-1:123456789012:benefit-allocation/benalloc-abc123def456"
  ],
  "detail": {
    "catalog": "AWS",
    "benefitAllocation": {
      "id": "benalloc-abc123def456",
      "fulfillmentType": "CREDITS",
      "status": "ACTIVE"
    },
    "benefitApplication": {
      "id": "benappl-12345abcdef123"
    },
    "benefit": {
      "id": "ben-xyz789uvw012",
      "name": "AWS Partner Credits",
      "program": ["AWS Partner Network"]
    }
  }
}
```

Alocação de benefícios inativada

Contexto de acionamento

O evento de alocação de benefícios inativada é publicado na Amazon EventBridge quando uma alocação de benefícios passa para o status INATIVO por meio do Serviço de benefícios. Esse evento ocorre quando um revisor interno atualiza o status da alocação por meio da UpdateBenefitAllocationInternal API ou quando uma alocação de benefícios expira automaticamente.

Esse evento geralmente ocorre quando uma alocação é temporariamente suspensa (devido à expiração, mudanças na política, problemas de conformidade ou motivos administrativos), tornando os benefícios indisponíveis para utilização pelo parceiro até serem reativados.

Destinatários

A AWS conta pertencente ao parceiro receberá a mensagem. Em outras palavras, a AWS Conta que possui o recurso.

Evento de exemplo

```
{
  "id": "4op55432-r7qo-k75n-p969-9mm8463t707u",
  "detail-type": "Benefit Allocation Inactivated",
  "source": "aws.partnercentral-benefits",
  "account": "123456789012",
  "time": "2025-06-30T23:59:59Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:partnercentral:us-east-1:123456789012:benefit-allocation/benalloc-abc123def456"
  ],
  "detail": {
    "catalog": "AWS",
    "benefitAllocation": {
      "id": "benalloc-abc123def456",
      "fulfillmentType": "CREDITS",
      "status": "INACTIVE"
    },
    "benefitApplication": {
      "id": "benappl-12345abcdef123"
    },
    "benefit": {
      "id": "ben-xyz789uvw012",
      "name": "AWS Partner Credits",
      "program": ["AWS Partner Network"]
    }
  }
}
```

Alocação de benefícios cumprida

Contexto de acionamento

O evento Benefit Allocation Fulfilled é publicado na Amazon EventBridge quando uma alocação de benefícios passa para o status FULFILLED por meio do Benefits Service. Esse evento ocorre quando

um revisor interno atualiza o status da alocação por meio da UpdateBenefitAllocationInternal API para indicar que a alocação foi totalmente utilizada ou que os termos do benefício foram atendidos.

Esse evento representa a conclusão do ciclo de vida do benefício, indicando que o parceiro utilizou com sucesso o benefício alocado ou que a alocação chegou à sua conclusão natural.

Destinatários

A AWS conta pertencente ao parceiro receberá a mensagem. Em outras palavras, a AWS Conta que possui o recurso.

Evento de exemplo

```
{
  "id": "5pq66543-s8rp-l86o-q070-0nn9574u818v",
  "detail-type": "Benefit Allocation Fulfilled",
  "source": "aws.partnercentral-benefits",
  "account": "123456789012",
  "time": "2025-12-15T17:30:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:partnercentral:us-east-1:123456789012:benefit-allocation/benalloc-abc123def456"
  ],
  "detail": {
    "catalog": "AWS",
    "benefitAllocation": {
      "id": "benalloc-abc123def456",
      "fulfillmentType": "CREDITS",
      "status": "FULFILLED"
    },
    "benefitApplication": {
      "id": "benappl-12345abcdef123"
    },
    "benefit": {
      "id": "ben-xyz789uvw012",
      "name": "AWS Partner Credits",
      "program": ["AWS Partner Network"]
    }
  }
}
```

Notificações para o AWS API do Partner Central Channel

Os eventos da API do canal fornecem notificações em tempo real sobre mudanças no status ou nos detalhes das atividades relacionadas ao canal. Esses eventos ajudam a manter seus sistemas sincronizados com o AWS Partner Central e ajudam a garantir respostas e atualizações oportunas.

Pré-requisitos

Você precisa das permissões apropriadas do IAM para acessar e gerenciar eventos da API do AWS Partner Central Channel. Para obter informações sobre as ações, os recursos e as chaves de condição disponíveis para EventBridge, consulte [Usando as condições da política do IAM na Amazon EventBridge](#) no Guia EventBridge do usuário da Amazon. Uma das chaves de condição é `events:detail-type`, que você pode usar para definir o escopo de permissões para tipos específicos de eventos.

O exemplo de política a seguir demonstra como personalizar e definir o escopo das permissões para os eventos. A `AllowPutRuleForPartnercentralChannelEvents` declaração permite a criação de regras, mas somente para eventos da `aws.partnercentral-channel` fonte.

Para exemplos detalhados de políticas do IAM, consulte a documentação da AWS sobre EventBridge permissões.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowPutRuleForPartnerCentralChannelEvents",
      "Effect": "Allow",
      "Action": "events:PutRule",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "events:source": "aws.partnercentral-channel"
        }
      }
    }
  ]
}
```

Configure EventBridge a Amazon para monitorar eventos

Para monitorar os eventos da API do canal, você cria uma EventBridge regra que corresponde aos eventos que você deseja capturar. Você pode usar o Console de gerenciamento da AWS ou os AWS SDKs para criar e gerenciar regras. As seções a seguir explicam como criar regras usando os dois métodos. Independentemente do método usado, você deve criar a regra na us-east-1 região Leste dos EUA (Norte da Virgínia).

Console de gerenciamento da AWS configuração

Para configurar uma EventBridge regra usando o Console de gerenciamento da AWS, siga as etapas em [Criação de regras que reagem a eventos na Amazon EventBridge](#) no Guia EventBridge do usuário da Amazon. Ao criar a regra, você deve definir o barramento de eventos como padrão e criar a regra na us-east-1 região Leste dos EUA (Norte da Virgínia).

Veja a seguir um exemplo de uma regra de evento:

```
{
  "source": ["aws.partnercentral-channel"],
  "detail": {
    "catalog": ["AWS"]
  }
}
```

AWS Configuração do SDK

Você pode usar os AWS SDKs para criar e gerenciar EventBridge regras de forma programática. Para obter mais informações, consulte [PutRule](#) a Amazon EventBridge API Reference.

O exemplo a seguir usa o AWS SDK para Python (Boto3):

```
import boto3

client = boto3.client('events', region_name='us-east-1')

response = client.put_rule(
    Name='ChannelHandshakeCreatedRule',
    EventPattern=
    '{
      "source": ["aws.partnercentral-channel"],
      "detail-type": ["Channel Handshake Created"],
```

```

        "detail": {"catalog": ["AWS"]}
    },
    State='ENABLED'
)
print('Rule ARN:', response['RuleArn'])

```

Saiba mais sobre os eventos da API do canal

As seções a seguir descrevem os tipos de eventos da API do canal, os cenários que os acionam e os exemplos de eventos.

Tipos de eventos

A seguir estão os tipos de eventos e seus gatilhos.

- Aperto de [mão de canal criado](#): acionado quando um novo aperto de mão de canal é criado.
- Aperto de [mão de canal aceito](#): acionado quando um novo aperto de mão de canal é aceito.
- Aperto de [mão do canal rejeitado](#): acionado quando um novo aperto de mão do canal é rejeitado.

Canal Handshake criado

Os exemplos a seguir mostram eventos típicos do Channel Handshake Created para diferentes tipos de handshake.

Inicie o período de serviço Handshake:

```

{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "Channel Handshake Created",
  "source": "aws.partnercentral-channel",
  "account": "789789789789",
  "time": "2025-02-01T00:00:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:partnercentral:us-east-1:123123123123:catalog/AWS/channel-handshake/ch-abc123def456g"
  ],
  "detail": {
    "requestId": "12345678-1234-1234-1234-123456789abc",
    "catalog": "AWS",

```

```

    "associatedResourceIdentifier": "rs-jkl012mno345p",
    "handshakeType": "START_SERVICE_PERIOD",
    "id": "ch-abc123def456g",
    "receiverAccountId": "789789789789",
    "ownerAccountId": "123123123123",
    "senderAccountId": "456456456456",
    "senderDisplayName": "TestDisplayName",
    "handshakeDetail": {
      "startServicePeriodHandshakeDetail": {
        "servicePeriodType": "MINIMUM_NOTICE_PERIOD",
        "minimumNoticeDays": "14",
        "endDate": null,
        "startDate": "2025-02-02T00:00:00Z",
        "note": "Test note example"
      }
    }
  }
}

```

Revogar o Handshake do Período de Serviço:

```

{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "Channel Handshake Created",
  "source": "aws.partnercentral-channel",
  "account": "789789789789",
  "time": "2025-02-01T00:00:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:partnercentral:us-east-1:123123123123:catalog/AWS/channel-handshake/ch-def456ghi789j"
  ],
  "detail": {
    "requestId": "12345678-1234-1234-1234-123456789abc",
    "catalog": "AWS",
    "associatedResourceIdentifier": "rs-jkl012mno345p",
    "handshakeType": "REVOKE_SERVICE_PERIOD",
    "id": "ch-def456ghi789j",
    "ownerAccountId": "123123123123",
    "receiverAccountId": "789789789789",
    "senderAccountId": "456456456456",
    "senderDisplayName": "TestDisplayName",
  }
}

```

```

    "handshakeDetail": {
      "revokeServicePeriodHandshakeDetail": {
        "servicePeriodType": "MINIMUM_NOTICE_PERIOD",
        "minimumNoticeDays": "14",
        "endDate": null,
        "startDate": "2025-02-02T00:00:00Z",
        "note": "Test note example"
      }
    }
  }
}

```

Handshake da conta de gerenciamento de programas:

```

{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "Channel Handshake Created",
  "source": "aws.partnercentral-channel",
  "account": "456456456456",
  "time": "2025-02-01T00:00:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:partnercentral:us-east-1:123123123123:catalog/AWS/channel-handshake/ch-ghi789jkl012m"
  ],
  "detail": {
    "requestId": "12345678-1234-1234-1234-123456789abc",
    "catalog": "AWS",
    "associatedResourceIdentifier": "pma-abc123def456g",
    "handshakeType": "PROGRAM_MANAGEMENT_ACCOUNT",
    "id": "ch-ghi789jkl012m",
    "ownerAccountId": "123123123123",
    "receiverAccountId": "456456456456",
    "senderAccountId": "123123123123",
    "senderDisplayName": "TestDisplayName",
    "handshakeDetail": {
      "programManagementAccountHandshakeDetail": {
        "program": "SOLUTION_PROVIDER"
      }
    }
  }
}

```

Handshake de canal aceito

Os exemplos a seguir mostram eventos típicos do Channel Handshake Accepted para diferentes tipos de handshake.

Inicie o período de serviço Handshake:

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "Channel Handshake Accepted",
  "source": "aws.partnercentral-channel",
  "account": "123123123123",
  "time": "2025-02-01T00:00:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:partnercentral:us-east-1:123123123123:catalog/AWS/channel-handshake/ch-abc123def456g"
  ],
  "detail": {
    "requestId": "12345678-1234-1234-1234-123456789abc",
    "catalog": "AWS",
    "associatedResourceIdentifier": "rs-jkl012mno345p",
    "handshakeType": "START_SERVICE_PERIOD",
    "id": "ch-abc123def456g",
    "ownerAccountId": "123123123123",
    "receiverAccountId": "789789789789",
    "senderAccountId": "456456456456",
    "senderDisplayName": "TestDisplayName",
    "handshakeDetail": {
      "startServicePeriodHandshakeDetail": {
        "servicePeriodType": "MINIMUM_NOTICE_PERIOD",
        "minimumNoticeDays": "14",
        "endDate": null,
        "startDate": "2025-02-02T00:00:00Z",
        "note": "Test note example"
      }
    }
  }
}
```

Revogar o Handshake do Período de Serviço:

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "Channel Handshake Accepted",
  "source": "aws.partnercentral-channel",
  "account": "123123123123",
  "time": "2025-02-01T00:00:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:partnercentral:us-east-1:123123123123:catalog/AWS/channel-handshake/ch-def456ghi789j"
  ],
  "detail": {
    "requestId": "12345678-1234-1234-1234-123456789abc",
    "catalog": "AWS",
    "associatedResourceIdentifier": "rs-jkl012mno345p",
    "handshakeType": "REVOKE_SERVICE_PERIOD",
    "id": "ch-def456ghi789j",
    "ownerAccountId": "123123123123",
    "receiverAccountId": "789789789789",
    "senderAccountId": "456456456456",
    "senderDisplayName": "TestDisplayName",
    "handshakeDetail": {
      "revokeServicePeriodHandshakeDetail": {
        "servicePeriodType": "MINIMUM_NOTICE_PERIOD",
        "minimumNoticeDays": "14",
        "endDate": null,
        "startDate": "2025-02-02T00:00:00Z",
        "note": "Test note example"
      }
    }
  }
}
```

Handshake da conta de gerenciamento de programas:

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "Channel Handshake Accepted",
  "source": "aws.partnercentral-channel",
  "account": "123123123123",
  "time": "2025-02-01T00:00:00Z",
```

```

    "region": "us-east-1",
    "resources": [
      "arn:aws:partnercentral:us-east-1:123123123123:catalog/AWS/channel-handshake/
ch-ghi789jkl012m"
    ],
    "detail": {
      "requestId": "12345678-1234-1234-1234-123456789abc",
      "catalog": "AWS",
      "associatedResourceIdentifier": "pma-abc123def456g",
      "handshakeType": "PROGRAM_MANAGEMENT_ACCOUNT",
      "id": "ch-ghi789jkl012m",
      "ownerAccountId": "123123123123",
      "receiverAccountId": "456456456456",
      "senderAccountId": "123123123123",
      "senderDisplayName": "TestDisplayName",
      "handshakeDetail": {
        "programManagementAccountHandshakeDetail": {
          "program": "SOLUTION_PROVIDER"
        }
      }
    }
  }
}

```

Canal Handshake rejeitado

Os exemplos a seguir mostram eventos típicos do Channel Handshake Rejected para diferentes tipos de handshake.

Inicie o período de serviço Handshake:

```

{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "Channel Handshake Rejected",
  "source": "aws.partnercentral-channel",
  "account": "123123123123",
  "time": "2025-02-01T00:00:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:partnercentral:us-east-1:123123123123:catalog/AWS/channel-handshake/
ch-abc123def456g"
  ],
  "detail": {

```

```

    "requestId": "12345678-1234-1234-1234-123456789abc",
    "catalog": "AWS",
    "associatedResourceIdentifier": "rs-jkl012mno345p",
    "handshakeType": "START_SERVICE_PERIOD",
    "id": "ch-abc123def456g",
    "ownerAccountId": "123123123123",
    "receiverAccountId": "789789789789",
    "senderAccountId": "456456456456",
    "senderDisplayName": "TestDisplayName",
    "handshakeDetail": {
      "startServicePeriodHandshakeDetail": {
        "servicePeriodType": "MINIMUM_NOTICE_PERIOD",
        "minimumNoticeDays": "14",
        "endDate": null,
        "startDate": "2025-02-02T00:00:00Z",
        "note": "Test note example"
      }
    }
  }
}

```

Revogar o Handshake do Período de Serviço:

```

{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "Channel Handshake Rejected",
  "source": "aws.partnercentral-channel",
  "account": "123123123123",
  "time": "2025-02-01T00:00:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:partnercentral:us-east-1:123123123123:catalog/AWS/channel-handshake/ch-def456ghi789j"
  ],
  "detail": {
    "requestId": "12345678-1234-1234-1234-123456789abc",
    "catalog": "AWS",
    "associatedResourceIdentifier": "rs-jkl012mno345p",
    "handshakeType": "REVOKE_SERVICE_PERIOD",
    "id": "ch-def456ghi789j",
    "ownerAccountId": "123123123123",
    "receiverAccountId": "789789789789",

```

```

    "senderAccountId": "456456456456",
    "senderDisplayName": "TestDisplayName",
    "handshakeDetail": {
      "revokeServicePeriodHandshakeDetail": {
        "servicePeriodType": "MINIMUM_NOTICE_PERIOD",
        "minimumNoticeDays": "14",
        "endDate": null,
        "startDate": "2025-02-02T00:00:00Z",
        "note": "Test note example"
      }
    }
  }
}

```

Handshake da conta de gerenciamento de programas:

```

{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "Channel Handshake Rejected",
  "source": "aws.partnercentral-channel",
  "account": "123123123123",
  "time": "2025-02-01T00:00:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:partnercentral:us-east-1:123123123123:catalog/AWS/channel-handshake/ch-ghi789jkl012m"
  ],
  "detail": {
    "requestId": "12345678-1234-1234-1234-123456789abc",
    "catalog": "AWS",
    "associatedResourceIdentifier": "pma-abc123def456g",
    "handshakeType": "PROGRAM_MANAGEMENT_ACCOUNT",
    "id": "ch-ghi789jkl012m",
    "ownerAccountId": "123123123123",
    "receiverAccountId": "456456456456",
    "senderAccountId": "123123123123",
    "senderDisplayName": "TestDisplayName",
    "handshakeDetail": {
      "programManagementAccountHandshakeDetail": {
        "program": "SOLUTION_PROVIDER"
      }
    }
  }
}

```

```
}  
}
```

Testando em uma sandbox

Os parceiros podem usar um sandbox para testar e validar suas interações com a API do AWS Partner Central em um ambiente seguro e isolado, garantindo uma operação tranquila antes de promover sua solução para o ambiente de produção. AWS oferece um sandbox dinâmico para usuários da API do AWS Partner Central que retorna respostas semelhantes às do ambiente de produção. AWS não fornece uma interface de usuário para o ambiente sandbox. Portanto, os parceiros precisam confiar nas respostas programáticas para testar suas soluções.

Acesso ao ambiente sandbox

Os parceiros obtêm acesso ao ambiente de teste assim que vinculam sua conta da AWS à AWS Partner Central. Para obter mais informações, consulte [Vinculando sua AWS conta à conta do AWS Partner Central](#). Cada solicitação inclui um parâmetro de catálogo que determina o ambiente de dados. Quando o catálogo é definido como AWS, ele faz referência aos dados de produção e, quando definido como Sandbox, faz referência aos dados do sandbox.

Detalhes importantes sobre o ambiente sandbox

1. Atualização de dados: uma vez por ano, AWS atualiza os dados no ambiente sandbox (normalmente no início do ano). Após essa atualização, você pode perder alguns dos dados em seu ambiente de teste.
2. Escopo do teste: o ambiente sandbox é normalmente usado para testes funcionais e não para testar escalabilidade ou desempenho.

Tópicos

- [Testando em uma caixa de areia para o AWS API de conta do Partner Central](#)
- [Testando em uma caixa de areia para o AWS API de vendas da Central de Parceiros](#)
- [Testando em uma caixa de areia para o AWS API de benefícios do Partner Central](#)
- [Testando em uma caixa de areia para o AWS API do Partner Central Channel](#)
- [Testando em uma caixa de areia para o AWS API de medição de receita do Partner Central](#)

Testando em uma caixa de areia para o AWS API de conta do Partner Central

Como usar o sandbox

Para usar o sandbox, conclua as seguintes etapas:

1. Crie uma função do IAM:

Crie uma função do IAM Conta da AWS vinculada à sua conta do AWS Partner Central.

2. Atribuir política:

Atribua a política a seguir à função do IAM. Para obter mais informações, consulte [Adicionar e remover permissões de identidade do IAM](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AccountManagement",
      "Effect": "Allow",
      "Action": [
        "partnercentral:AcceptConnectionInvitation",
        "partnercentral:AssociateAwsTrainingCertificationEmailDomain",
        "partnercentral:CancelConnection",
        "partnercentral:CancelConnectionInvitation",
        "partnercentral:CancelProfileUpdateTask",
        "partnercentral:CreateConnectionInvitation",
        "partnercentral:CreatePartner",
        "partnercentral:DisassociateAwsTrainingCertificationEmailDomain",
        "partnercentral:GetAllianceLeadContact",
        "partnercentral:GetConnection",
        "partnercentral:GetConnectionInvitation",
        "partnercentral:GetConnectionPreferences",
        "partnercentral:GetPartner",
        "partnercentral:GetProfileUpdateTask",
        "partnercentral:GetProfileVisibility",
        "partnercentral:GetVerification",
        "partnercentral:ListConnectionInvitations",
        "partnercentral:ListConnections",
        "partnercentral:ListPartners",
        "partnercentral:PutAllianceLeadContact",
```

```

        "partnercentral:PutProfileVisibility",
        "partnercentral:RejectConnectionInvitation",
        "partnercentral:SendEmailVerificationCode",
        "partnercentral:StartProfileUpdateTask",
"partnercentral:StartVerification",
        "partnercentral:UpdateConnectionPreferences"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "partnercentral:Catalog": [
                "Sandbox"
            ]
        }
    }
},
{
    "Sid": "TaggingAccess",
    "Effect": "Allow",
    "Action": [
        "partnercentral:TagResource",
        "partnercentral:UntagResource",
        "partnercentral:ListTagsForResource"
    ],
    "Resource": [
        "arn:aws:partnercentral:*:*:catalog/Sandbox/partner/*"
    ],
    "Condition": {
        "StringEquals": {
            "partnercentral:Catalog": [
                "Sandbox"
            ]
        }
    }
}
]
}

```

3. Use as credenciais da função do IAM:

Use as credenciais (chave secreta e chave de acesso) dessa função do IAM em sua solução para realizar as ações da API.

Teste AWS actions

Durante a fase de teste, muitas vezes é necessário simular AWS ações. Essa simulação permite que os parceiros testem minuciosamente o fluxo completo de sua integração com AWS os serviços. Cada solicitação inclui um parâmetro de catálogo, que determina o ambiente de dados.

Simulando a criação de um parceiro

Para simular a criação de um parceiro, na carga útil da `CreatePartner` ação, inclua `"catalog": "Sandbox"` na carga útil.

Por exemplo:

```
{
  "ClientToken": "client-generated-uuid",
  "Catalog": "Sandbox",
  "LegalName": "Your Name",
  "PrimarySolutionType": "SOLUTION_TYPE",
  "AllianceLeadContact": {
    "FirstName": "Example First Name",
    "LastName": "Example Last Name",
    "BusinessTitle": "Example Title",
    "Email": "example@domain.com"
  },
  "EmailVerificationCode": "123456"
}
```

Nota: No Sandbox, a verificação de e-mail está desativada. Você pode usar qualquer um de seis dígitos `EmailVerificationCode` para testar. Além disso, não use “Tags” na solicitação.

Notas de teste adicionais

1. Para testar outras ações, defina `"catalog": "Sandbox"` a carga útil.
2. Para testar as conexões da conta do parceiro no sandbox, você deve executar a `StartProfileUpdateTask` ação após criar o parceiro no catálogo do sandbox.
3. Para passar para a produção, altere o valor do catálogo de Sandbox para AWS.

Eventos de teste no ambiente sandbox

Os parceiros podem consumir eventos do ambiente sandbox para ajudar a testar as implementações baseadas em eventos. Configure `EventBridge` o mesmo Conta da AWS com regras para ouvir

eventos de sandbox especificando `catalog`: Sandbox nos detalhes do evento. Para obter mais informações, consulte [Notificações para o AWS API de conta do Partner Central](#).

Exemplo de regra de evento:

```
{
  "source": ["aws.partnercentral-account"],
  "detail": {
    "catalog": ["Sandbox"]
  }
}
```

As regras de eventos que especificam `catalog` como Sandbox corresponderão somente aos eventos provenientes do sandbox, gerados pelas ações que você executa no ambiente sandbox.

Testando em uma caixa de areia para o AWS API de vendas da Central de Parceiros

Como usar o sandbox

1. Crie uma função do IAM:

Crie uma função do IAM Conta da AWS vinculada à sua conta do AWS Partner Central.

2. Atribuir política:

Atribua a política a seguir à função do IAM. Para obter mais informações, consulte [Adicionar e remover permissões de identidade do IAM](#).

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "partnercentral:CreateOpportunity",
        "partnercentral:UpdateOpportunity",
        "partnercentral:ListOpportunities",
        "partnercentral:GetOpportunity",
        "partnercentral:GetAwsOpportunitySummary",
        "partnercentral:ListSolutions",

```

```

        "partnercentral:AssociateOpportunity",
        "partnercentral:DisassociateOpportunity",
        "partnercentral:AssignOpportunity",
        "partnercentral:SubmitOpportunity",
        "partnercentral:AcceptEngagementInvitation",
        "partnercentral:CreateEngagementInvitation",
        "partnercentral:RejectEngagementInvitation",
        "partnercentral:GetEngagementInvitation",
        "partnercentral:ListEngagementInvitations",
        "partnercentral:StartEngagementFromOpportunityTask",
        "partnercentral:StartEngagementByAcceptingInvitationTask",
        "partnercentral:CreateResourceSnapshotJob",
        "partnercentral:StartResourceSnapshotJob",
        "partnercentral:CreateEngagement"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "partnercentral:Catalog": "Sandbox"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "aws-marketplace:ListEntities",
        "aws-marketplace:DescribeEntity"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "aws-marketplace:SearchAgreements",
        "aws-marketplace:DescribeAgreement"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws-marketplace:PartyType": "Proposer"
        }
    }
}
]

```

```
}
```

3. Use as credenciais da função do IAM:

Use as credenciais (chave secreta e chave de acesso) dessa função do IAM em sua solução para realizar as ações da API.

Teste AWS actions

Durante a fase de teste, muitas vezes é necessário simular AWS ações. Essa simulação permite que os parceiros testem minuciosamente o fluxo completo de sua integração com AWS os serviços.

Simulando a criação de um AWS oportunidade originada

Para simular a criação de uma oportunidade AWS originada, na carga útil da CreateOpportunity ação, inclua e. "Catalog": "Sandbox" "Origin": "AWS Referral"

Por exemplo:

```
{
  "Catalog": "Sandbox",
  "Origin": "AWS Referral",
  "OpportunityIdentifier": "0123456",
  "Title": "Test Opportunity",
  ...
}
```

Simulando atualizações em uma oportunidade originada por um parceiro

Para simular atualizações ou outras ações em uma oportunidade originada pelo parceiro, use a UpdateOpportunity ação com Lifecycle.ReviewStatus: "Approved" e/ou "Action Required" na carga "Catalog": "Sandbox" útil.

Se você estiver retirando a carga da GetOpportunity ação para fazer a atualização, certifique-se de alterar a ID para Identifier e remover os seguintes campos:

- CreatedDate
- OpportunityTeam
- RelatedEntityIdentifiers

Por exemplo:

```
{
  "Catalog": "Sandbox",
  "Identifier": "0123456",
  "Title": "Updated Test Opportunity",
  "Lifecycle": {
    "ReviewStatus": "Approved"
  }
  ...
}
```

Eventos de teste no ambiente sandbox

Os parceiros podem consumir eventos de oportunidade do ambiente sandbox para ajudar a testar as implementações baseadas em eventos. Configure EventBridge o mesmo Conta da AWS com regras para ouvir eventos de sandbox especificando `catalog: sandbox` nos detalhes do evento. Para obter mais informações, consulte [Notificações para o AWS API de vendas da Central de Parceiros](#).

Exemplo de regra de evento:

```
{
  "source": ["aws.partnercentral-selling"],
  "detail": {
    "catalog": ["Sandbox"]
  }
}
```

As regras de eventos que especificam `catalog` como `Sandbox` corresponderão somente aos eventos provenientes do sandbox, gerados pelas ações que você executa no ambiente sandbox.

Notas de teste adicionais

1. AssociateOpportunityAção de teste:

- Use a solução padrão "S-1234567" para testar a AssociateOpportunity ação.
- Para testar as ofertas do Marketplace, use um ID de oferta real da sua conta.

2. Para passar para a produção, altere o `catalog` valor de `Sandbox` para `AWS`.

Como obter ajuda

Se você encontrar desafios para integrar seu CRM ou precisar testar um cenário específico não abordado aqui, entre em contato com AWS o suporte apresentando um caso por meio das seguintes etapas:

1. Faça login no [AWS Partner Central](#) com suas credenciais do AWS Partner Network.
2. No [Support Center do Partner Central](#), escolha Open New Case registrar um novo caso. Preencha os campos da seguinte forma:
 - a. Tipo de estojo de suporte: AWS Partner Central.
 - b. Pergunta sobre: Partner Central Tools or ACE leads and opportunities.
 - c. Seja específico: selecione o tipo de caso de integração de CRM mais adequado.
 - d. Assunto: Inclua uma breve descrição da solicitação.
 - e. Descrição: forneça uma descrição detalhada dos problemas, perguntas, erros e etapas de solução de problemas.
 - f. Anexos: inclua registros e capturas de tela, quando aplicável.

Testando em uma caixa de areia para o AWS API de benefícios do Partner Central

Como usar o sandbox

Para usar o sandbox, conclua as seguintes etapas:

1. Crie uma função do IAM:

Crie uma função do IAM Conta da AWS vinculada à sua conta do AWS Partner Central.

2. Atribuir política:

Atribua a política a seguir à função do IAM. Para obter mais informações, consulte [Adicionar e remover permissões de identidade do IAM](#).

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "BenefitsManagement",
    "Effect": "Allow",
```

```

    "Action": [
        "partnercentral:ListBenefits",
        "partnercentral:GetBenefit",
        "partnercentral:CreateBenefitApplication",
        "partnercentral:AmendBenefitApplication",
        "partnercentral:UpdateBenefitApplication",
        "partnercentral:SubmitBenefitApplication",
        "partnercentral:GetBenefitApplication",
        "partnercentral:CancelBenefitApplication",
        "partnercentral:RecallBenefitApplication",
        "partnercentral:ListBenefitApplications",
        "partnercentral:AssociateBenefitApplicationResource",
        "partnercentral:DisassociateBenefitApplicationResource",
        "partnercentral:ListBenefitAllocations",
        "partnercentral:GetBenefitAllocation",
        "partnercentral:TagResource",
        "partnercentral:UntagResource",
        "partnercentral:ListTagsForResource"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "partnercentral:Catalog": [
                "Sandbox"
            ]
        }
    }
},
{
    "Sid": "AWSPartnerOpportunityAccess",
    "Effect": "Allow",
    "Action": [
        "partnercentral:GetAwsOpportunitySummary",
        "partnercentral:GetOpportunity",
        "partnercentral:ListOpportunities"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "partnercentral:Catalog": [
                "Sandbox"
            ]
        }
    }
}

```

```

    },
    {
      "Sid": "ListingAWSMarketplaceEntities",
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:ListEntities"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AWSMarketplaceOffersAccess",
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:DescribeEntity"
      ],
      "Resource": [
        "arn:aws:aws-marketplace::AWSMarketplace/Offer/*"
      ]
    },
    {
      "Sid": "S3PutObjectAccess",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject"
      ],
      "Resource": ["arn:aws:s3:::aws-partner-central-marketplace-ephemeral-
writeonly-files/*"]
    },
    {
      "Sid": "TaggingAccess",
      "Effect": "Allow",
      "Action": [
        "partnercentral:TagResource",
        "partnercentral:UntagResource",
        "partnercentral:ListTagsForResource"
      ],
      "Resource": [
        "arn:aws:partnercentral:us-east-1:*:catalog/Sandbox/benefit-
application/*",
        "arn:aws:partnercentral:us-east-1:*:catalog/Sandbox/benefit-
allocation/*"
      ],
      "Condition": {

```

```

        "StringEquals": {
            "partnercentral:Catalog": [
                "Sandbox"
            ]
        }
    }
}

```

3. Use as credenciais da função do IAM:

Use as credenciais (chave secreta e chave de acesso) dessa função do IAM em sua solução para realizar as ações da API.

Teste AWS actions

Durante a fase de teste, muitas vezes é necessário simular AWS ações. Essa simulação permite que os parceiros testem minuciosamente o fluxo completo de sua integração com AWS os serviços. Cada solicitação inclui um parâmetro de catálogo, que determina o ambiente de dados.

Simulando a criação de um aplicativo de benefícios

Para simular a criação de um aplicativo de benefícios, na carga útil da `CreateBenefitApplication` ação, inclua `"catalog": "Sandbox"` na carga útil.

Por exemplo:

```

{
  "Catalog": "Sandbox",
  "ClientToken": "String",
  "BenefitIdentifier": "String",
  "FulfillmentTypes": ["String"],
  "BenefitApplicationDetails": JsonDocument,
  "Name": "String",
  "Description": "String",
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ],

```

```
"PartnerContacts": [
  {
    "BusinessTitle": "string",
    "Email": "string",
    "FirstName": "string",
    "LastName": "string",
    "Phone": "string"
  }
],
"FileDetails": [
  {
    "FileURI": "String",
    "BusinessUseCase": "String"
  }
],
"AssociatedResources": [
  {
    "ResourceType": "BENEFIT_ALLOCATION | OPPORTUNITY",
    "ResourceArn": "ARN"
  }
]
}
```

Notas de teste adicionais

1. Para testar outras ações, defina "catalog": "Sandbox" a carga útil.
2. Os IDs de benefícios no catálogo Sandbox e no catálogo da AWS são diferentes. Faça uma chamada de GetBenefit API com "catalog": "Sandbox" para obter o Identificador de Benefícios na Sandbox.
3. A alocação de benefícios no catálogo Sandbox e no catálogo da AWS é diferente.
4. Para passar para a produção, altere o valor do catálogo de Sandbox para AWS.

Eventos de teste no ambiente sandbox

Os parceiros podem consumir eventos do ambiente sandbox para ajudar a testar as implementações baseadas em eventos. Configure EventBridge o mesmo Conta da AWS com regras para ouvir eventos de sandbox especificando catalog: Sandbox nos detalhes do evento. Para obter mais informações, consulte [Notificações para o AWS API de benefícios do Partner Central](#).

Exemplo de regra de evento:

```
{
  "source": ["aws.partnercentral-benefits"],
  "detail": {
    "catalog": ["Sandbox"]
  }
}
```

As regras de eventos que especificam `catalog` como `Sandbox` corresponderão somente aos eventos provenientes do sandbox, gerados pelas ações que você executa no ambiente sandbox.

Testando em uma caixa de areia para o AWS API do Partner Central Channel

As contas no sandbox não estão qualificadas para os benefícios do canal e não são validadas de acordo com todos os requisitos do programa.

Como usar o sandbox

Para usar o sandbox, conclua as seguintes etapas:

1. Crie uma função do IAM:

Crie uma função do IAM Conta da AWS vinculada à sua conta do AWS Partner Central.

2. Atribuir política:

Atribua a política a seguir à função do IAM. Para obter mais informações, consulte [Adicionar e remover permissões de identidade do IAM](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ChannelManagement",
      "Effect": "Allow",
      "Action": [
        "partnercentral:CreateProgramManagementAccount",
        "partnercentral:UpdateProgramManagementAccount",
        "partnercentral>DeleteProgramManagementAccount",
        "partnercentral:ListProgramManagementAccounts",
        "partnercentral:GetProgramManagementAccount",

```

```

        "partnercentral:CreateRelationship",
        "partnercentral:UpdateRelationship",
        "partnercentral:DeleteRelationship",
        "partnercentral:GetRelationship",
        "partnercentral:ListRelationships",
        "partnercentral:CreateChannelHandshake",
        "partnercentral:AcceptChannelHandshake",
        "partnercentral:RejectChannelHandshake",
        "partnercentral:CancelChannelHandshake",
        "partnercentral:ListChannelHandshakes"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "partnercentral:Catalog": [
                "Sandbox"
            ]
        }
    }
},
{
    "Sid": "TaggingAccess",
    "Effect": "Allow",
    "Action": [
        "partnercentral:TagResource",
        "partnercentral:UntagResource",
        "partnercentral:ListTagsForResource"
    ],
    "Resource": [
        "arn:aws:partnercentral:*:*:catalog/Sandbox/program-management-account/*",
        "arn:aws:partnercentral:*:*:catalog/Sandbox/channel-handshake/*"
    ],
    "Condition": {
        "StringEquals": {
            "partnercentral:Catalog": [
                "Sandbox"
            ]
        }
    }
}
]
}

```

3. Use as credenciais da função do IAM:

Use as credenciais (chave secreta e chave de acesso) dessa função do IAM em sua solução para realizar as ações da API.

Teste AWS actions

Durante a fase de teste, muitas vezes é necessário simular AWS ações. Essa simulação permite que os parceiros testem minuciosamente o fluxo completo de sua integração com AWS os serviços. Cada solicitação inclui um parâmetro de catálogo, que determina o ambiente de dados.

Simulando a criação de uma conta de gerenciamento de programas

Para simular a criação de uma Conta de Gerenciamento do Programa, na carga útil da `CreateProgramManagementAccount` ação, inclua "catalog": "Sandbox" na carga útil.

Por exemplo:

```
{
  "catalog": "Sandbox",
  "accountId": "123456789012",
  "displayName": "Test PMA Name",
  "clientToken": "123abc456def789ghi",
  "program": "SOLUTION_PROVIDER",
  "tags": [
    {
      "key": "testkey",
      "value": "testvalue"
    }
  ]
}
```

Simulando atualizações em um relacionamento

Para simular atualizações em um relacionamento, use a `UpdateRelationship` ação com "catalog": "Sandbox" na carga útil.

Por exemplo:

```
{
  "catalog": "Sandbox",
  "displayName": "Updated Test PMA",
  "identifier": "rs-abc123def456g",
```

```
"programManagementAccountIdentifier": "pma-123abc456def7"
}
```

Notas de teste adicionais

1. Para testar outras ações, defina "catalog": "Sandbox" a carga útil.
2. Para passar para a produção, altere o valor do catálogo de Sandbox para AWS.

Eventos de teste no ambiente sandbox

Os parceiros podem consumir eventos do ambiente sandbox para ajudar a testar as implementações baseadas em eventos. Configure EventBridge o mesmo Conta da AWS com regras para ouvir eventos de sandbox especificando catalog: Sandbox nos detalhes do evento. Para obter mais informações, consulte [Notificações para o AWS API do Partner Central Channel](#).

Exemplo de regra de evento:

```
{
  "source": ["aws.partnercentral-channel"],
  "detail": {
    "catalog": ["Sandbox"]
  }
}
```

As regras de eventos que especificam catalog como Sandbox corresponderão somente aos eventos provenientes do sandbox, gerados pelas ações que você executa no ambiente sandbox.

Como obter ajuda

Se você encontrar desafios nos testes ou precisar testar um cenário específico não abordado aqui, entre em contato com o suporte apresentando um caso por meio das seguintes etapas:

1. Faça login no [AWS Partner Central](#) com suas credenciais do AWS Partner Network.
2. No [Support Center do Partner Central](#), escolha Open New Case registrar um novo caso. Preencha os campos da seguinte forma:
 - a. Tipo de estojo de suporte: Channel Program
 - b. Pergunta sobre: General Program Inquiry
 - c. Seja específico: selecione o tipo de caso de canal mais adequado.

- d. Assunto: Inclua uma breve descrição da solicitação.
- e. Descrição: forneça uma descrição detalhada dos problemas, perguntas, erros e etapas de solução de problemas.
- f. Anexos: inclua registros e capturas de tela, quando aplicável.

Testando em uma caixa de areia para o AWS API de medição de receita do Partner Central

Como usar o sandbox

Para usar o sandbox, conclua as seguintes etapas:

1. Crie uma função do IAM:

Crie uma função do IAM Conta da AWS vinculada à sua conta do AWS Partner Central.

2. Atribuir política:

Atribua a política a seguir à função do IAM. Para obter mais informações, consulte [Adicionar e remover permissões de identidade do IAM](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RevenueMeasurementCreation",
      "Effect": "Allow",
      "Action": [
        "partnercentral:CreateRevenueAttribution",
        "partnercentral:CreateMarketplaceRevenueShare",
        "partnercentral:CreateMarketplaceRevenueShareAllocation"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "partnercentral:Catalog": [
            "Sandbox"
          ]
        }
      }
    }
  ],
}
```

```

{
  "Sid": "RevenueMeasurementListing",
  "Effect": "Allow",
  "Action": [
    "partnercentral:ListRevenueAttributions",
    "partnercentral:ListRevenueAttributionAllocations",
    "partnercentral:ListMarketplaceRevenueShares",
    "partnercentral:ListMarketplaceRevenueShareAllocations"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "partnercentral:Catalog": [
        "Sandbox"
      ]
    }
  }
},
{
  "Sid": "RevenueAttributionManagement",
  "Effect": "Allow",
  "Action": [
    "partnercentral:GetRevenueAttribution",
    "partnercentral:UpdateRevenueAttribution",
    "partnercentral:GetRevenueAttributionAllocation",
    "partnercentral:StartRevenueAttributionAllocationsTask",
    "partnercentral:GetRevenueAttributionAllocationsTask"
  ],
  "Resource": "arn:aws:partnercentral:*:*:catalog/Sandbox/revenue-attribution/*",
  "Condition": {
    "StringEquals": {
      "partnercentral:Catalog": [
        "Sandbox"
      ]
    }
  }
},
{
  "Sid": "MarketplaceRevenueShareManagement",
  "Effect": "Allow",
  "Action": [
    "partnercentral:GetMarketplaceRevenueShare",
    "partnercentral:GetMarketplaceRevenueShareAllocation",

```

```

        "partnercentral:UpdateMarketplaceRevenueShareAllocation"
    ],
    "Resource": "arn:aws:partnercentral:*:*:catalog/Sandbox/marketplace-
revenue-share/*",
    "Condition": {
        "StringEquals": {
            "partnercentral:Catalog": [
                "Sandbox"
            ]
        }
    }
},
{
    "Sid": "TaggingAccess",
    "Effect": "Allow",
    "Action": [
        "partnercentral:TagResource",
        "partnercentral:UntagResource",
        "partnercentral:ListTagsForResource"
    ],
    "Resource": [
        "arn:aws:partnercentral:*:*:catalog/Sandbox/revenue-attribution/*",
        "arn:aws:partnercentral:*:*:catalog/Sandbox/marketplace-revenue-
share/*"
    ],
    "Condition": {
        "StringEquals": {
            "partnercentral:Catalog": [
                "Sandbox"
            ]
        }
    }
},
{
    "Sid": "OpportunityAccess",
    "Effect": "Allow",
    "Action": [
        "partnercentral:ListOpportunities",
        "partnercentral:GetOpportunity"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "partnercentral:Catalog": [

```

```

        "Sandbox"
    ]
}
},
{
    "Sid": "ListingAWSMarketplaceEntities",
    "Effect": "Allow",
    "Action": [
        "aws-marketplace:ListEntities"
    ],
    "Resource": "*"
},
{
    "Sid": "AWSMarketplaceEntityAccess",
    "Effect": "Allow",
    "Action": [
        "aws-marketplace:DescribeEntity"
    ],
    "Resource": [
        "arn:aws:aws-marketplace:*:*:AWSMarketplace*/Offer/*"
    ]
}
]
}

```

3. Use as credenciais da função do IAM:

Use as credenciais (chave secreta e chave de acesso) dessa função do IAM em sua solução para realizar as ações da API.

Teste AWS actions

Durante a fase de teste, muitas vezes é necessário simular AWS ações. Essa simulação permite que os parceiros testem minuciosamente o fluxo completo de sua integração com AWS os serviços. Cada solicitação inclui um parâmetro de catálogo, que determina o ambiente de dados.

Simulando a criação de um ID de atribuição de receita

Para simular a criação de um ID de atribuição de receita, na carga útil da `CreateRevenueAttribution` ação, `"Catalog": "Sandbox"` inclua na carga útil.

Por exemplo:

```
{
  "Catalog": "Sandbox",
  "ClientToken": "String",
  "Name": "String",
  "Description": "String",
  "TenancyModel": "MULTI_TENANT | SINGLE_TENANT",
  "ProductIdentifier": "String",
  "Tags": [
    {
      "Key": "String",
      "Value": "String"
    }
  ]
}
```

Notas de teste adicionais

1. Para testar outras ações, defina "Catalog": "Sandbox" a carga útil.
2. Para passar para a produção, altere o valor do catálogo de Sandbox para AWS.

Exemplos de código para o Partner Central usando AWS SDKs

Os exemplos de código a seguir mostram como usar o Partner Central com um kit AWS de desenvolvimento de software (SDK).

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar perfis de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cenários são exemplos de código que mostram como realizar tarefas específicas chamando várias funções dentro de um serviço ou combinadas com outros Serviços da AWS.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Utilizar AWS API do Partner Central com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Exemplos de código

- [Exemplos básicos de uso do Partner Central AWS SDKs](#)
 - [Ações para o Partner Central usando AWS SDKs](#)
 - [Use AssignOpportunitycom um AWS SDK](#)
 - [Use AssociateOpportunitycom um AWS SDK](#)
 - [Use CreateOpportunitycom um AWS SDK](#)
 - [Use DisassociateOpportunitycom um AWS SDK](#)
 - [Use GetAwsOpportunitySummarycom um AWS SDK](#)
 - [Use GetEngagementInvitationcom um AWS SDK](#)
 - [Use GetOpportunitycom um AWS SDK](#)
 - [Use ListEngagementInvitationscom um AWS SDK](#)
 - [Use ListOpportunitiescom um AWS SDK](#)
 - [Use ListSolutionscom um AWS SDK](#)
 - [Use RejectEngagementInvitationcom um AWS SDK](#)
 - [Use StartEngagementByAcceptingInvitationTaskcom um AWS SDK](#)
 - [Use StartEngagementFromOpportunityTaskcom um AWS SDK](#)
 - [Use UpdateOpportunitycom um AWS SDK](#)

- [Cenários para o uso do Partner Central AWS SDKs](#)
 - [Atualizar a entidade associada de uma oportunidade](#)

Exemplos básicos de uso do Partner Central AWS SDKs

Os exemplos de código a seguir mostram como usar os conceitos básicos do AWS Partner Central com AWS SDKs.

Exemplos

- [Ações para o Partner Central usando AWS SDKs](#)
 - [Use AssignOpportunitycom um AWS SDK](#)
 - [Use AssociateOpportunitycom um AWS SDK](#)
 - [Use CreateOpportunitycom um AWS SDK](#)
 - [Use DisassociateOpportunitycom um AWS SDK](#)
 - [Use GetAwsOpportunitySummarycom um AWS SDK](#)
 - [Use GetEngagementInvitationcom um AWS SDK](#)
 - [Use GetOpportunitycom um AWS SDK](#)
 - [Use ListEngagementInvitationscom um AWS SDK](#)
 - [Use ListOpportunitiescom um AWS SDK](#)
 - [Use ListSolutionscom um AWS SDK](#)
 - [Use RejectEngagementInvitationcom um AWS SDK](#)
 - [Use StartEngagementByAcceptingInvitationTaskcom um AWS SDK](#)
 - [Use StartEngagementFromOpportunityTaskcom um AWS SDK](#)
 - [Use UpdateOpportunitycom um AWS SDK](#)

Ações para o Partner Central usando AWS SDKs

Os exemplos de código a seguir demonstram como realizar ações individuais do Partner Central com AWS SDKs. Cada exemplo inclui um link para GitHub, onde você pode encontrar instruções para configurar e executar o código.

Esses trechos chamam a API da Central de Parceiros e são trechos de código de programas maiores que devem ser executados no contexto. É possível ver as ações em contexto em [Cenários para o uso do Partner Central AWS SDKs](#).

Os exemplos a seguir incluem apenas as ações mais utilizadas. Consulte uma lista completa na [Referência de API da Central de Parceiros da AWS](#).

Exemplos

- [Use AssignOpportunitycom um AWS SDK](#)
- [Use AssociateOpportunitycom um AWS SDK](#)
- [Use CreateOpportunitycom um AWS SDK](#)
- [Use DisassociateOpportunitycom um AWS SDK](#)
- [Use GetAwsOpportunitySummarycom um AWS SDK](#)
- [Use GetEngagementInvitationcom um AWS SDK](#)
- [Use GetOpportunitycom um AWS SDK](#)
- [Use ListEngagementInvitationscom um AWS SDK](#)
- [Use ListOpportunitiescom um AWS SDK](#)
- [Use ListSolutionscom um AWS SDK](#)
- [Use RejectEngagementInvitationcom um AWS SDK](#)
- [Use StartEngagementByAcceptingInvitationTaskcom um AWS SDK](#)
- [Use StartEngagementFromOpportunityTaskcom um AWS SDK](#)
- [Use UpdateOpportunitycom um AWS SDK](#)

Use **AssignOpportunity**com um AWS SDK

Os exemplos de código a seguir mostram como usar o AssignOpportunity.

Java

SDK para Java 2.x

Reatribua uma oportunidade existente a outro usuário.

```
package org.example;

import static org.example.utils.Constants.*;

import org.example.utils.Constants;
import org.example.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
```

```
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.regions.Region;
import
    software.amazon.awssdk.services.partnercentralselling.PartnerCentralSellingClient;
import
    software.amazon.awssdk.services.partnercentralselling.model.AssignOpportunityRequest;
import
    software.amazon.awssdk.services.partnercentralselling.model.AssignOpportunityResponse;
import
    software.amazon.awssdk.services.partnercentralselling.model.AssigneeContact;

/*
Purpose
PC-API-07 Assigning a new owner
*/

public class AssignOpportunity {

    static PartnerCentralSellingClient client =
        PartnerCentralSellingClient.builder()
            .region(Region.US_EAST_1)
            .credentialsProvider(DefaultCredentialsProvider.create())
            .httpClient(ApacheHttpClient.builder().build())
            .build();

    public static void main(String[] args) {

        String opportunityId = args.length > 0 ? args[0] : OPPORTUNITY_ID;

        String assigneeFirstName = "John";

        String assigneeLastName = "Doe";

        String assigneeEmail = "test@test.com";

        String businessTitle = "PartnerAccountManager";

        AssignOpportunityResponse response = getResponse(opportunityId,
            assigneeFirstName, assigneeLastName, assigneeEmail, businessTitle);

        ReferenceCodesUtils.formatOutput(response);
    }
}
```

```

static AssignOpportunityResponse getResponse(String opportunityId, String
assigneeFirstName, String assigneeLastName, String assigneeEmail, String
businessTitle) {

    AssignOpportunityRequest assignOpportunityRequest =
AssignOpportunityRequest.builder()
        .catalog(Constants.CATALOG_TO_USE)
        .identifier(opportunityId)
        .assignee(AssigneeContact.builder()
            .firstName(assigneeFirstName)
            .lastName(assigneeLastName)
            .email(assigneeEmail)
            .businessTitle(businessTitle)
            .build())
        .build();

    AssignOpportunityResponse response =
client.assignOpportunity(assignOpportunityRequest);

    return response;
}
}

```

- Para obter detalhes da API, consulte [AssignOpportunity](#) a Referência AWS SDK for Java 2.x da API.

Python

SDK para Python (Boto3)

Reatribua uma oportunidade existente a outro usuário.

```

#!/usr/bin/env python

"""
Purpose
PC-API-07 Assigning a new owner
"""

import logging
import boto3
import utils.helpers as helper

```

```
from botocore.client import ClientError

from utils.constants import CATALOG_TO_USE

serviceName = "partnercentral-selling"

partner_central_client = boto3.client(
    service_name=serviceName,
    region_name='us-east-1'
)

def assign_opportunity(identifier):
    assign_opportunity_request = {
        "Catalog": CATALOG_TO_USE,
        "Identifier": identifier,
        "Assignee": {
            "BusinessTitle": "OpportunityOwner",
            "Email": "test@test.com",
            "FirstName": "John",
            "LastName": "Doe"
        }
    }
    try:
        # Perform an API call
        response =
partner_central_client.assign_opportunity(**assign_opportunity_request)
        return response

    except ClientError as err:
        # Catch all client exceptions
        print(err.response)

def usage_demo():
    identifier = "04236468"

    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Assigning a new owner to an opportunity.")
    print("-" * 88)

    helper.pretty_print_datetime(assign_opportunity(identifier))

if __name__ == "__main__":
```

```
usage_demo()
```

- Para obter detalhes da API, consulte a [AssignOpportunity](#) Referência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Utilizar AWS API do Partner Central com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **AssociateOpportunity** com um AWS SDK

Os exemplos de código a seguir mostram como usar o AssociateOpportunity.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- [Atualizar a entidade associada de uma oportunidade](#)

Java

SDK para Java 2.x

Crie uma associação formal entre uma oportunidade e várias entidades relacionadas.

```
package org.example;

import static org.example.utils.Constants.*;

import org.example.utils.Constants;
import org.example.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.regions.Region;
import
    software.amazon.awssdk.services.partnercentralselling.PartnerCentralSellingClient;
import
    software.amazon.awssdk.services.partnercentralselling.model.AssociateOpportunityRequest;
import
    software.amazon.awssdk.services.partnercentralselling.model.AssociateOpportunityResponse;
```

```
/*
Purpose
PC-API -11 Associating a product
PC-API -12 Associating a solution
PC-API -13 Associating an offer
entity_type = Solutions | AWSProducts | AWSMarketplaceOffers
*/

public class AssociateOpportunity {

    static PartnerCentralSellingClient client =
        PartnerCentralSellingClient.builder()
            .region(Region.US_EAST_1)
            .credentialsProvider(DefaultCredentialsProvider.create())
            .httpClient(ApacheHttpClient.builder().build())
            .build();

    public static void main(String[] args) {

        String opportunityId = args.length > 0 ? args[0] : OPPORTUNITY_ID;

        String entityType = "Solutions";

        String entityIdIdentifier = "S-00000000";

        AssociateOpportunityResponse response = getResponse(opportunityId,
            entityType, entityIdIdentifier );

        ReferenceCodesUtils.formatOutput(response);
    }

    static AssociateOpportunityResponse getResponse(String opportunityId, String
        entityType, String entityIdIdentifier) {

        AssociateOpportunityRequest associateOpportunityRequest =
        AssociateOpportunityRequest.builder()
            .catalog(Constants.CATALOG_TO_USE)
            .opportunityIdentifier(opportunityId)
            .relatedEntityType(entityType)
            .relatedEntityIdentifier(entityIdIdentifier)
            .build();

        AssociateOpportunityResponse response =
        client.associateOpportunity(associateOpportunityRequest);
    }
}
```

```

        return response;
    }
}

```

- Para obter detalhes da API, consulte [AssociateOpportunity](#) na Referência AWS SDK for Java 2.x da API.

Python

SDK para Python (Boto3)

Crie uma associação formal entre uma oportunidade e várias entidades relacionadas.

```

#!/usr/bin/env python

"""
Purpose
PC-API -11 Associating a product
PC-API -12 Associating a solution
PC-API -13 Associating an offer
"""

import logging
import boto3
import utils.helpers as helper
from botocore.client import ClientError

from utils.constants import CATALOG_TO_USE

serviceName = "partnercentral-selling"

partner_central_client = boto3.client(
    service_name=serviceName,
    region_name='us-east-1'
)

def associate_opportunity(entity_type, entity_identifier, opportunityIdentifier):
    associate_opportunity_request = {
        "Catalog": CATALOG_TO_USE,
        "OpportunityIdentifier" : opportunityIdentifier,
        "RelatedEntityType" : entity_type,

```

```

        "RelatedEntityIdentifier" : entity_identifier
    }
    try:
        # Perform an API call
        response =
partner_central_client.associate_opportunity(**associate_opportunity_request)
        return response

    except ClientError as err:
        # Catch all client exceptions
        print(err.response)

def usage_demo():
    #entity_type = Solutions | AWSProducts | AWSMarketplaceOffers
    entity_type = "Solutions"
    entity_identifier = "S-0059717"
    opportunityIdentifier = "05465588"

    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Associate Opportunity.")
    print("-" * 88)

    helper.pretty_print_datetime(associate_opportunity(entity_type,
    entity_identifier, opportunityIdentifier))

if __name__ == "__main__":
    usage_demo()

```

- Para obter detalhes da API, consulte a [AssociateOpportunity](#)Referência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Utilizar AWS API do Partner Central com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **CreateOpportunity** com um AWS SDK

Os exemplos de código a seguir mostram como usar o CreateOpportunity.

.NET

SDK para .NET

Crie uma oportunidade.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// PDX-License-Identifier: Apache-2.0

using System;
using Newtonsoft.Json;
using Amazon;
using Amazon.Runtime;
using Amazon.PartnerCentralSelling;
using Amazon.PartnerCentralSelling.Model;

namespace AWSExample
{
    class Program
    {
        static readonly string catalogToUse = "AWS";
        static async Task Main(string[] args)
        {
            // Initialize credentials from .aws/credentials file
            var credentials = new
            Amazon.Runtime.CredentialManagement.SharedCredentialsFile();
            if (credentials.TryGetProfile("default", out var profile))
            {
                AWSCredentials awsCredentials =
            profile.GetAWSCredentials(credentials);

                var client = new
            AmazonPartnerCentralSellingClient(awsCredentials);

                var request = new CreateOpportunityRequest
                {
                    Catalog = catalogToUse,
                    Origin = "Partner Referral",
                    Customer = new Customer
                    {
                        Account = new Account
                        {
                            Address = new Address
                            {

```

```

        CountryCode = "US",
        PostalCode = "99502",
        StateOrRegion = "Alaska"
    },
    CompanyName = "TestCompanyName",
    Duns = "123456789",
    WebsiteUrl = "www.test.io",
    Industry = "Automotive"
},
Contacts = new List<Contact>
{
    new Contact
    {
        Email = "test@test.io",
        FirstName = "John ",
        LastName = "Doe",
        Phone = "+144444444444",
        BusinessTitle = "test title"
    }
},
LifeCycle = new LifeCycle
{
    ReviewStatus = "Submitted",
    TargetCloseDate = "2024-12-30"
},
Marketing = new Marketing
{
    Source = "None"
},
OpportunityType = "Net New Business",
PrimaryNeedsFromAws = new List<string> { "Co-Sell -
Architectural Validation" },
Project = new Project
{
    Title = "Moin Test UUID",
    CustomerBusinessProblem = "Sandbox is not working as
expected",

    CustomerUseCase = "AI Machine Learning and Analytics",
    DeliveryModels = new List<string> { "SaaS or PaaS" },
    ExpectedCustomerSpend = new List<ExpectedCustomerSpend>
    {
        new ExpectedCustomerSpend
        {

```

```

        Amount = "2000.0",
        CurrencyCode = "USD",
        Frequency = "Monthly",
        TargetCompany = "Ibexlabs"
    }
},
    SalesActivities = new List<string> { "Initialized
discussions with customer" }
}
};

try
{
    var response = await client.CreateOpportunityAsync(request);
    Console.WriteLine(response.HttpStatusCode);
    string formattedJson = JsonConvert.SerializeObject(response,
Formatting.Indented);
    Console.WriteLine(formattedJson);
}
catch (ValidationException ex)
{
    Console.WriteLine("Validation error: " + ex.Message);
}
catch (AmazonPartnerCentralSellingException e)
{
    Console.WriteLine("Failed:");
    Console.WriteLine(e.RequestId);
    Console.WriteLine(e.ErrorCode);
    Console.WriteLine(e.Message);
}
}
else
{
    Console.WriteLine("Profile not found.");
}
}
}
}
}

```

- Para obter detalhes da API, consulte [CreateOpportunity](#) a Referência AWS SDK para .NET da API.

Java

SDK para Java 2.x

Crie uma oportunidade.

```
package org.example;

import java.time.Instant;
import java.util.ArrayList;
import java.util.List;

import static org.example.utils.Constants.*;

import org.example.entity.Root;
import org.example.utils.ReferenceCodesUtils;
import org.example.utils.StringSerializer;

import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.regions.Region;
import
    software.amazon.awssdk.services.partnercentralselling.PartnerCentralSellingClient;
import software.amazon.awssdk.services.partnercentralselling.model.Account;
import software.amazon.awssdk.services.partnercentralselling.model.Address;
import software.amazon.awssdk.services.partnercentralselling.model.Contact;
import
    software.amazon.awssdk.services.partnercentralselling.model.CreateOpportunityRequest;
import
    software.amazon.awssdk.services.partnercentralselling.model.CreateOpportunityResponse;
import software.amazon.awssdk.services.partnercentralselling.model.Customer;
import
    software.amazon.awssdk.services.partnercentralselling.model.ExpectedCustomerSpend;
import software.amazon.awssdk.services.partnercentralselling.model.LifeCycle;
import software.amazon.awssdk.services.partnercentralselling.model.Marketing;
import software.amazon.awssdk.services.partnercentralselling.model.MonetaryValue;
import
    software.amazon.awssdk.services.partnercentralselling.model.NextStepsHistory;
import software.amazon.awssdk.services.partnercentralselling.model.Project;
import
    software.amazon.awssdk.services.partnercentralselling.model.SoftwareRevenue;

import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
```

```
import com.google.gson.ToNumberPolicy;

public class CreateOpportunity {

    static final Gson GSON = new GsonBuilder()
        .setObjectToNumberStrategy(ToNumberPolicy.LAZILY_PARSED_NUMBER)
        .registerTypeAdapter(String.class, new StringSerializer())
        .create();

    static PartnerCentralSellingClient client =
        PartnerCentralSellingClient.builder()
            .region(Region.US_EAST_1)
            .credentialsProvider(DefaultCredentialsProvider.create())
            .httpClient(ApacheHttpClient.builder().build())
            .build();

    public static void main(String[] args) {

        String inputFile = "CreateOpportunity2.json";

        if (args.length > 0)
            inputFile = args[0];

        CreateOpportunityResponse response = createOpportunity(inputFile);

        client.close();
    }

    static CreateOpportunityResponse createOpportunity(String inputFile) {

        String inputString = ReferenceCodesUtils.readInputFileToString(inputFile);

        Root root = GSON.fromJson(inputString, Root.class);

        List<NextStepsHistory> nextStepsHistories = new ArrayList<NextStepsHistory>();
        if ( root.lifeCycle != null && root.lifeCycle.nextStepsHistories != null) {
            for (org.example.entity.NextStepsHistory nextStepsHistoryJson :
                root.lifeCycle.nextStepsHistories) {
                NextStepsHistory nextStepsHistory = NextStepsHistory.builder()
                    .time(Instant.parse(nextStepsHistoryJson.time))
                    .value(nextStepsHistoryJson.value)
                    .build();
                nextStepsHistories.add(nextStepsHistory);
            }
        }
    }
}
```

```
}

Lifecycle lifeCycle = null;
if ( root.lifeCycle != null ) {
    lifeCycle = Lifecycle.builder()
        .closedLostReason(root.lifeCycle.closedLostReason)
        .nextSteps(root.lifeCycle.nextSteps)
        .nextStepsHistory(nextStepsHistories)
        .reviewComments(root.lifeCycle.reviewComments)
        .reviewStatus(root.lifeCycle.reviewStatus)
        .reviewStatusReason(root.lifeCycle.reviewStatusReason)
        .stage(root.lifeCycle.stage)
        .targetCloseDate(root.lifeCycle.targetCloseDate)
        .build();
}

Marketing marketing = null;
if ( root.marketing != null ) {
    marketing = Marketing.builder()
        .awsFundingUsed(root.marketing.awsFundingUsed)
        .campaignName(root.marketing.campaignName)
        .channels(root.marketing.channels)
        .source(root.marketing.source)
        .useCases(root.marketing.useCases)
        .build();
}

Address address = null;
if ( root.customer != null && root.customer.account != null &&
    root.customer.account.address != null ) {
    address = Address.builder()
        .city(root.customer.account.address.city)
            .postalCode(root.customer.account.address.postalCode)
            .stateOrRegion(root.customer.account.address.stateOrRegion)
            .countryCode(root.customer.account.address.countryCode)
            .streetAddress(root.customer.account.address.streetAddress)
            .build();
}

Account account = null;
if ( root.customer != null && root.customer.account != null ) {
    account = Account.builder()
        .address(address)
```

```

        .awsAccountId(root.customer.account.awsAccountId)
        .duns(root.customer.account.duns)
        .industry(root.customer.account.industry)
        .otherIndustry(root.customer.account.otherIndustry)
        .companyName(root.customer.account.companyName)
        .websiteUrl(root.customer.account.websiteUrl)
        .build();
    }

    List<Contact> contacts = new ArrayList<Contact>();
    if ( root.customer != null && root.customer.contacts != null) {
        for (org.example.entity.Contact jsonContact : root.customer.contacts) {
            Contact contact = Contact.builder()
                .email(jsonContact.email)
                .firstName(jsonContact.firstName)
                .lastName(jsonContact.lastName)
                .phone(jsonContact.phone)
                .businessTitle(jsonContact.businessTitle)
                .build();
            contacts.add(contact);
        }
    }

    Customer customer = Customer.builder()
        .account(account)
        .contacts(contacts)
        .build();

    Contact opportunityTeamContact = null;
    if (root.opportunityTeam != null && root.opportunityTeam.get(0) != null ) {
        opportunityTeamContact = Contact.builder()
            .firstName(root.opportunityTeam.get(0).firstName)
            .lastName(root.opportunityTeam.get(0).lastName)
            .email(root.opportunityTeam.get(0).email)
            .phone(root.opportunityTeam.get(0).phone)
            .businessTitle(root.opportunityTeam.get(0).businessTitle)
            .build();
    }

    List<ExpectedCustomerSpend> expectedCustomerSpend = new
    ArrayList<ExpectedCustomerSpend>();
    if ( root.project != null && root.project.expectedCustomerSpend != null) {
        for (org.example.entity.ExpectedCustomerSpend expectedCustomerSpendJson :
        root.project.expectedCustomerSpend) {

```

```

ExpectedCustomerSpend expectedCustomerSpend = null;
expectedCustomerSpend = ExpectedCustomerSpend.builder()
    .amount(expectedCustomerSpendJson.amount)
    .currencyCode(expectedCustomerSpendJson.currencyCode)
    .frequency(expectedCustomerSpendJson.frequency)
    .targetCompany(expectedCustomerSpendJson.targetCompany)
    .build();
expectedCustomerSpend.add(expectedCustomerSpend);
}

}

Project project = null;
if ( root.project != null) {
    project = Project.builder()
        .title(root.project.title)
        .customerBusinessProblem(root.project.customerBusinessProblem)
        .customerUseCase(root.project.customerUseCase)
        .deliveryModels(root.project.deliveryModels)
        .expectedCustomerSpend(expectedCustomerSpend)
        .salesActivities(root.project.salesActivities)
        .competitorName(root.project.competitorName)
        .otherSolutionDescription(root.project.otherSolutionDescription)
        .build();
}

SoftwareRevenue softwareRevenue = null;
if ( root.softwareRevenue != null) {
    MonetaryValue monetaryValue = null;
    if ( root.softwareRevenue.value != null) {
        monetaryValue = MonetaryValue.builder()
            .amount(root.softwareRevenue.value.amount)
            .currencyCode(root.softwareRevenue.value.currencyCode)
            .build();
    }
    softwareRevenue = SoftwareRevenue.builder()
        .deliveryModel(root.softwareRevenue.deliveryModel)
        .effectiveDate(root.softwareRevenue.effectiveDate)
        .expirationDate(root.softwareRevenue.expirationDate)
        .value(monetaryValue)
        .build();
}

// Building the Actual CreateOpportunity Request

```

```
CreateOpportunityRequest createOpportunityRequest =
CreateOpportunityRequest.builder()
    .catalog(CATALOG_TO_USE)
    .clientToken(root.clientToken)
    .primaryNeedsFromAwsWithStrings(root.primaryNeedsFromAws)
    .opportunityType(root.opportunityType)
    .lifeCycle(lifeCycle)
    .marketing(marketing)
    .nationalSecurity(root.nationalSecurity)
    .origin(root.origin)
    .customer(customer)
    .project(project)
    .partnerOpportunityIdentifier(root.partnerOpportunityIdentifier)
    .opportunityTeam(opportunityTeamContact)
    .softwareRevenue(softwareRevenue)
    .build();

CreateOpportunityResponse response =
client.createOpportunity(createOpportunityRequest);
System.out.println("Successfully created: " + response);

return response;
}
}
```

- Para obter detalhes da API, consulte [CreateOpportunity](#) a Referência AWS SDK for Java 2.x da API.

Python

SDK para Python (Boto3)

Crie uma oportunidade.

```
#!/usr/bin/env python
import boto3
import logging
import sys
import os
sys.path.append(os.path.dirname(os.path.dirname(os.path.abspath(__file__))))
```

```
import utils.helpers as helper
import utils.stringify_details as sd
from botocore.client import ClientError
from utils.constants import CATALOG_TO_USE

serviceName = "partnercentral-selling"

def create_opportunity(partner_central_client):
    create_opportunity_request = helper.remove_nulls(sd.stringify_json("src/
create_opportunity/createOpportunity.json"))
    try:
        # Perform an API call
        response =
partner_central_client.create_opportunity(**create_opportunity_request)

        helper.pretty_print_datetime(response)

        # Retrieve the opportunity details
        get_response = partner_central_client.get_opportunity(
            Identifier=response["Id"],
            Catalog=CATALOG_TO_USE
        )
        helper.pretty_print_datetime(get_response)
        return response
    except ClientError as err:
        # Catch all client exceptions
        print(err.response)

def usage_demo():
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Create Opportunity.")
    print("-" * 88)

    partner_central_client = boto3.client(
        service_name=serviceName,
        region_name='us-east-1'
    )

    create_opportunity(partner_central_client)

if __name__ == "__main__":
```

```
usage_demo()
```

- Para obter detalhes da API, consulte a [CreateOpportunity](#) Referência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Utilizar AWS API do Partner Central com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **DisassociateOpportunity** com um AWS SDK

Os exemplos de código a seguir mostram como usar o `DisassociateOpportunity`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- [Atualizar a entidade associada de uma oportunidade](#)

Java

SDK para Java 2.x

Remova uma associação existente entre uma oportunidade e entidades relacionadas.

```
package org.example;

import static org.example.utils.Constants.*;

import org.example.utils.Constants;
import org.example.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.regions.Region;
import
    software.amazon.awssdk.services.partnercentralselling.PartnerCentralSellingClient;
import
    software.amazon.awssdk.services.partnercentralselling.model.DisassociateOpportunityRequest;
import
    software.amazon.awssdk.services.partnercentralselling.model.DisassociateOpportunityResponse;

/*
```

```

Purpose
PC-API -14 Removing a Solution
PC-API -15 Removing an offer
PC-API -16 Removing a product
entity_type = Solutions | AWSProducts | AWSMarketplaceOffers
*/

public class DisassociateOpportunity {

    static PartnerCentralSellingClient client =
    PartnerCentralSellingClient.builder()
        .region(Region.US_EAST_1)
        .credentialsProvider(DefaultCredentialsProvider.create())
        .httpClient(ApacheHttpClient.builder().build())
        .build();

    public static void main(String[] args) {

        String opportunityId = args.length > 0 ? args[0] : OPPORTUNITY_ID;

        String entityType = "Solutions";

        String entityIdIdentifier = "S-00000000";

        DisassociateOpportunityResponse response = getResponse(opportunityId,
        entityType, entityIdIdentifier );

        ReferenceCodesUtils.formatOutput(response);
    }

    static DisassociateOpportunityResponse getResponse(String opportunityId, String
    entityType, String entityIdIdentifier) {

        DisassociateOpportunityRequest disassociateOpportunityRequest =
        DisassociateOpportunityRequest.builder()
            .catalog(Constants.CATALOG_T0_USE)
            .opportunityIdentifier(opportunityId)
            .relatedEntityType(entityType)
            .relatedEntityIdentifier(entityIdIdentifier)
            .build();

        DisassociateOpportunityResponse response =
        client.disassociateOpportunity(disassociateOpportunityRequest);
    }
}

```

```
        return response;
    }
}
```

- Para obter detalhes da API, consulte [DisassociateOpportunity](#) a Referência AWS SDK for Java 2.x da API.

Python

SDK para Python (Boto3)

Remova uma associação existente entre uma oportunidade e entidades relacionadas.

```
#!/usr/bin/env python

"""
Purpose
PC-API -14 Removing a Solution
PC-API -15 Removing an offer
PC-API -16 Removing a product
"""

import logging
import boto3
import utils.helpers as helper
from botocore.client import ClientError

from utils.constants import CATALOG_TO_USE

serviceName = "partnercentral-selling"

partner_central_client = boto3.client(
    service_name=serviceName,
    region_name='us-east-1'
)

def disassociate_opportunity(entity_type, entity_identifier,
    opportunityIdentifier):
    disassociate_opportunity_request = {
        "Catalog": CATALOG_TO_USE,
        "OpportunityIdentifier" : opportunityIdentifier,
        "RelatedEntityType" : entity_type,
```

```

        "RelatedEntityIdentifier" : entity_identifier
    }
    try:
        # Perform an API call
        response =
partner_central_client.disassociate_opportunity(**disassociate_opportunity_request)
        return response

    except ClientError as err:
        # Catch all client exceptions
        print(err.response)

def usage_demo():
    #entity_type = Solutions | AWSProducts | AWSMarketplaceOffers
    entity_type = "Solutions"
    entity_identifier = "S-0049999"
    opportunityIdentifier = "04397574"

    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Get updated Opportunity.")
    print("-" * 88)

    helper.pretty_print_datetime(disassociate_opportunity(entity_type,
    entity_identifier, opportunityIdentifier))

if __name__ == "__main__":
    usage_demo()

```

- Para obter detalhes da API, consulte a [DisassociateOpportunity](#) Referência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Utilizar AWS API do Partner Central com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **GetAwsOpportunitySummary** com um AWS SDK

Os exemplos de código a seguir mostram como usar o **GetAwsOpportunitySummary**.

Java

SDK para Java 2.x

Recupera um resumo de uma AWS oportunidade.

```
package org.example;

import static org.example.utils.Constants.*;

import org.example.utils.Constants;
import org.example.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.regions.Region;
import
    software.amazon.awssdk.services.partnercentralselling.PartnerCentralSellingClient;
import
    software.amazon.awssdk.services.partnercentralselling.model.GetAwsOpportunitySummaryRequest;
import
    software.amazon.awssdk.services.partnercentralselling.model.GetAwsOpportunitySummaryResponse;

/**
 * Purpose
 * PC-API-25 Retrieves a summary of an AWS Opportunity.
 */

public class GetAwsOpportunitySummary {

    static PartnerCentralSellingClient client =
        PartnerCentralSellingClient.builder()
            .region(Region.US_EAST_1)
            .credentialsProvider(DefaultCredentialsProvider.create())
            .httpClient(ApacheHttpClient.builder().build())
            .build();

    public static void main(String[] args) {

        String opportunityId = args.length > 0 ? args[0] : OPPORTUNITY_ID;

        GetAwsOpportunitySummaryResponse response = getResponse(opportunityId);

        ReferenceCodesUtils.formatOutput(response);
    }
}
```

```

    }

    public static GetAwsOpportunitySummaryResponse getResponse(String opportunityId)
    {

        GetAwsOpportunitySummaryRequest getOpportunityRequest =
        GetAwsOpportunitySummaryRequest.builder()
            .catalog(Constants.CATALOG_TO_USE)
            .relatedOpportunityIdentifier(opportunityId)
            .build();

        GetAwsOpportunitySummaryResponse response =
        client.getAwsOpportunitySummary(getOpportunityRequest);

        return response;
    }
}

```

- Para obter detalhes da API, consulte [GetAwsOpportunitySummary](#) a Referência AWS SDK for Java 2.x da API.

Python

SDK para Python (Boto3)

Recupera um resumo de uma AWS oportunidade.

```

#!/usr/bin/env python

"""
Purpose
PC-API-25 Retrieves a summary of an AWS Opportunity.
    LifeCycle.ReviewStatus=Approved
"""

import logging
import boto3
import utils.helpers as helper
from botocore.client import ClientError

from utils.constants import CATALOG_TO_USE

```

```
serviceName = "partnercentral-selling"

partner_central_client = boto3.client(
    service_name=serviceName,
    region_name='us-east-1'
)

def get_opportunity(identifier):
    get_opportunity_request = {
        "Catalog": CATALOG_TO_USE,
        "RelatedOpportunityIdentifier": identifier
    }
    try:
        # Perform an API call
        response =
partner_central_client.get_aws_opportunity_summary(**get_opportunity_request)
        return response

    except ClientError as err:
        # Catch all client exceptions
        print(err.response)

def usage_demo():
    identifier = "05465588"

    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Get AWS Opportunity summary.")
    print("-" * 88)

    helper.pretty_print_datetime(get_opportunity(identifier))

if __name__ == "__main__":
    usage_demo()
```

- Para obter detalhes da API, consulte a [GetAwsOpportunitySummary](#) Referência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Utilizar AWS API do Partner Central com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **GetEngagementInvitation** com um AWS SDK

Os exemplos de código a seguir mostram como usar o `GetEngagementInvitation`.

Java

SDK para Java 2.x

Recupera os detalhes de um convite de engajamento compartilhado AWS por um parceiro.

```
package org.example;

import static org.example.utils.Constants.*;

import org.example.utils.Constants;
import org.example.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.regions.Region;
import
    software.amazon.awssdk.services.partnercentralselling.PartnerCentralSellingClient;
import
    software.amazon.awssdk.services.partnercentralselling.model.GetEngagementInvitationRequest;
import
    software.amazon.awssdk.services.partnercentralselling.model.GetEngagementInvitationResponse;

/*
 * Purpose
 * PC-API-22 Get engagement invitation opportunity
 */

public class GetEngagementInvitation {

    static PartnerCentralSellingClient client =
        PartnerCentralSellingClient.builder()
            .region(Region.US_EAST_1)
            .credentialsProvider(DefaultCredentialsProvider.create())
            .httpClient(ApacheHttpClient.builder().build())
```

```

        .build();

    public static void main(String[] args) {

        String opportunityId = args.length > 0 ? args[0] : OPPORTUNITY_ID;

        GetEngagementInvitationResponse response = getResponse(opportunityId);

        ReferenceCodesUtils.formatOutput(response);
    }

    static GetEngagementInvitationResponse getResponse(String opportunityId) {

        GetEngagementInvitationRequest getOpportunityRequest =
        GetEngagementInvitationRequest.builder()
            .catalog(Constants.CATALOG_TO_USE)
            .identifier(opportunityId)
            .build();

        GetEngagementInvitationResponse response =
        client.getEngagementInvitation(getOpportunityRequest);

        return response;
    }
}

```

- Para obter detalhes da API, consulte [GetEngagementInvitation](#) a Referência AWS SDK for Java 2.x da API.

Python

SDK para Python (Boto3)

Recupera os detalhes de um convite de engajamento compartilhado AWS por um parceiro.

```

#!/usr/bin/env python

"""
Purpose
PC-API-22  GetOpportunityEngagementInvitation - Retrieves details of a specific
engagement invitation.

```

```

This operation allows partners to view the invitation and its associated
information,
such as the customer, project, and lifecycle details.
"""
import json
import logging
import boto3
import utils.helpers as helper

from utils.constants import CATALOG_TO_USE

serviceName = "partnercentral-selling"

partner_central_client = boto3.client(
    service_name=serviceName,
    region_name='us-east-1'
)

def get_opportunity_engagement_invitation(identifier):
    get_opportunity_engagement_invitation_request = {
        "Catalog": CATALOG_TO_USE,
        "Identifier": identifier
    }
    try:
        # Perform an API call
        response =
partner_central_client.get_engagement_invitation(**get_opportunity_engagement_invitation
        return response

    except Exception as err:
        # Catch all client exceptions
        print(json.dumps(err.response))

def usage_demo():
    identifier = "arn:aws:partnercentral-selling:us-east-1:aws:catalog/Sandbox/
engagement-invitation/engi-00000000IS0Qga"

    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Given the ARN identifier, retrieve details of Opportunity Engagement
Invitation.")
    print("-" * 88)

```

```
helper.pretty_print_datetime(get_opportunity_engagement_invitation(identifier))

if __name__ == "__main__":
    usage_demo()
```

- Para obter detalhes da API, consulte a [GetEngagementInvitation](#) Referência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Utilizar AWS API do Partner Central com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **GetOpportunity** com um AWS SDK

Os exemplos de código a seguir mostram como usar o `GetOpportunity`.

.NET

SDK para .NET

Tenha uma oportunidade.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// PDX-License-Identifier: Apache-2.0

using System;
using Newtonsoft.Json;
using Amazon;
using Amazon.Runtime;
using Amazon.PartnerCentralSelling;
using Amazon.PartnerCentralSelling.Model;

namespace AWSExample
{
    class Program
    {
        static readonly string catalogToUse = "AWS";
        static readonly string identifier = "01111111";
        static async Task Main(string[] args)
        {
```

```

        // Initialize credentials from .aws/credentials file
        var credentials = new
Amazon.Runtime.CredentialManagement.SharedCredentialsFile();
        if (credentials.TryGetProfile("default", out var profile))
        {
            AWSCredentials awsCredentials =
profile.GetAWSCredentials(credentials);

            var client = new
AmazonPartnerCentralSellingClient(awsCredentials);

            var request = new GetOpportunityRequest
            {
                Catalog = catalogToUse,
                Identifier = identifier
            };

            try {
                var response = await client.GetOpportunityAsync(request);
                Console.WriteLine(response.HttpStatusCode);
                string formattedJson = JsonConvert.SerializeObject(response,
Formatting.Indented);
                Console.WriteLine(formattedJson);
            } catch (ValidationException ex) {
                Console.WriteLine("Validation error: " + ex.Message);
            } catch (AmazonPartnerCentralSellingException e) {
                Console.WriteLine("Failed:");
                Console.WriteLine(e.RequestId);
                Console.WriteLine(e.ErrorCode);
                Console.WriteLine(e.Message);
            }
        }
        else
        {
            Console.WriteLine("Profile not found.");
        }
    }
}
}

```

- Para obter detalhes da API, consulte [GetOpportunity](#) a Referência AWS SDK para .NET da API.

Go

SDK para Go V2

Tenha uma oportunidade.

```
package main

import (
    "context"
    "encoding/json"
    "fmt"
    "log"

    "github.com/aws/aws-sdk-go-v2/aws"
    "github.com/aws/aws-sdk-go-v2/config"
    "github.com/aws/aws-sdk-go-v2/service/partnercentralselling"
)

func main() {
    config, err := config.LoadDefaultConfig(context.TODO())

    if err != nil {
        log.Fatal(err)
    }

    config.Region = "us-east-1"

    client := partnercentralselling.NewFromConfig(config)

    output, err := client.GetOpportunity(context.TODO(),
        &partnercentralselling.GetOpportunityInput{
            Identifier: aws.String("01111111"),
            Catalog:   aws.String("AWS"),
        })

    if err != nil {
        log.Fatal(err)
    }
    log.Println("printing opportuniy...\n")

    jsonOutput, err := json.MarshalIndent(output, "", "    ")
}
```

```
    fmt.Println(string(jsonOutput))
}
```

- Para obter detalhes da API, consulte [GetOpportunity](#) a Referência AWS SDK para Go da API.

Java

SDK para Java 2.x

Tenha uma oportunidade.

```
package org.example;

import static org.example.utils.Constants.*;

import org.example.utils.Constants;
import org.example.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.regions.Region;
import
    software.amazon.awssdk.services.partnercentralselling.PartnerCentralSellingClient;
import
    software.amazon.awssdk.services.partnercentralselling.model.GetOpportunityRequest;
import
    software.amazon.awssdk.services.partnercentralselling.model.GetOpportunityResponse;

/*
 * Purpose
 * PC-API-08 Get updated Opportunity
 */

public class GetOpportunity {

    static PartnerCentralSellingClient client =
        PartnerCentralSellingClient.builder()
            .region(Region.US_EAST_1)
            .credentialsProvider(DefaultCredentialsProvider.create())
            .httpClient(ApacheHttpClient.builder().build())
            .build();
```

```

public static void main(String[] args) {

    String opportunityId = args.length > 0 ? args[0] : OPPORTUNITY_ID;

    GetOpportunityResponse response = getResponse(opportunityId);

    ReferenceCodesUtils.formatOutput(response);
}

public static GetOpportunityResponse getResponse(String opportunityId) {

    GetOpportunityRequest getOpportunityRequest =
    GetOpportunityRequest.builder()
        .catalog(Constants.CATALOG_TO_USE)
        .identifier(opportunityId)
        .build();

    GetOpportunityResponse response =
    client.getOpportunity(getOpportunityRequest);

    return response;
}
}

```

- Para obter detalhes da API, consulte [GetOpportunity](#) a Referência AWS SDK for Java 2.x da API.

Python

SDK para Python (Boto3)

Tenha uma oportunidade.

```

#!/usr/bin/env python

"""
Purpose
PC-API -08 Get updated Opportunity given opportunity id
"""
import logging

```

```
import boto3
import utils.helpers as helper
from botocore.client import ClientError

from utils.constants import CATALOG_TO_USE

serviceName = "partnercentral-selling"

partner_central_client = boto3.client(
    service_name=serviceName,
    region_name='us-east-1'
)

def get_opportunity(identifier):
    get_opportunity_request = {
        "Catalog": CATALOG_TO_USE,
        "Identifier": identifier
    }
    try:
        # Perform an API call
        response =
partner_central_client.get_opportunity(**get_opportunity_request)
        return response

    except ClientError as err:
        # Catch all client exceptions
        print(err.response)

def usage_demo():
    identifier = "05465588"

    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Get updated Opportunity.")
    print("-" * 88)

    helper.pretty_print_datetime(get_opportunity(identifier))

if __name__ == "__main__":
    usage_demo()
```

- Para obter detalhes da API, consulte a [GetOpportunity](#)Referência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Utilizar AWS API do Partner Central com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **ListEngagementInvitations** com um AWS SDK

Os exemplos de código a seguir mostram como usar o ListEngagementInvitations.

Java

SDK para Java 2.x

Recupera uma lista de convites de engajamento enviados ao parceiro.

```
package org.example;

import java.util.ArrayList;
import java.util.List;

import org.example.utils.ReferenceCodesUtils;
import static org.example.utils.Constants.*;

import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.regions.Region;
import
    software.amazon.awssdk.services.partnercentralselling.PartnerCentralSellingClient;
import
    software.amazon.awssdk.services.partnercentralselling.model.ListEngagementInvitationsReq
import
    software.amazon.awssdk.services.partnercentralselling.model.ListEngagementInvitationsRes
import
    software.amazon.awssdk.services.partnercentralselling.model.ParticipantType;
import
    software.amazon.awssdk.services.partnercentralselling.model.EngagementInvitationSummary;

public class ListEngagementInvitations {
```

```
static PartnerCentralSellingClient client =
PartnerCentralSellingClient.builder()
    .region(Region.US_EAST_1)
    .credentialsProvider(DefaultCredentialsProvider.create())
    .httpClient(ApacheHttpClient.builder().build())
    .build();

public static void main(String[] args) {

    List<EngagementInvitationSummary> opportunitySummaries = getResponse();
    ReferenceCodesUtils.formatOutput(opportunitySummaries);
}

static List<EngagementInvitationSummary> getResponse() {

    List<EngagementInvitationSummary> opportunitySummaries = new
    ArrayList<EngagementInvitationSummary>();

    ListEngagementInvitationsRequest listOpportunityRequest =
    ListEngagementInvitationsRequest.builder()
        .catalog(CATALOG_TO_USE)
        .participantType(ParticipantType.RECEIVER)
        .maxResults(5)
        .build();

    ListEngagementInvitationsResponse response =
    client.listEngagementInvitations(listOpportunityRequest);

    opportunitySummaries.addAll(response.engagementInvitationSummaries());

    client.close();

    return opportunitySummaries;
}
}
```

- Para obter detalhes da API, consulte [ListEngagementInvitations](#) a Referência AWS SDK for Java 2.x da API.

Python

SDK para Python (Boto3)

Recupera uma lista de convites de engajamento enviados ao parceiro.

```
#!/usr/bin/env python

"""
Purpose
PC-API-21 ListEngagementInvitations - Retrieves a list of engagement invitations
based on specified criteria.
This operation allows partners to view all invitations to engagement.
"""

import json
import logging
import boto3
import utils.helpers as helper

from utils.constants import CATALOG_TO_USE

serviceName = "partnercentral-selling"

partner_central_client = boto3.client(
    service_name=serviceName,
    region_name='us-east-1'
)

def list_engagement_invitations():
    list_engagement_invitations_request = {
        "Catalog": CATALOG_TO_USE,
        "MaxResults": 20
    }
    try:
        # Perform an API call
        response =
partner_central_client.list_engagement_invitations(**list_engagement_invitations_request)
        return response

    except Exception as err:
        # Catch all client exceptions
        print(json.dumps(err.response))
```

```
def usage_demo():
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Retrieve list of Engagement Invitations.")
    print("-" * 88)

    helper.pretty_print_datetime(list_engagement_invitations())

if __name__ == "__main__":
    usage_demo()
```

- Para obter detalhes da API, consulte a [ListEngagementInvitations](#) Referência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Utilizar AWS API do Partner Central com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **ListOpportunities** com um AWS SDK

Os exemplos de código a seguir mostram como usar o `ListOpportunities`.

.NET

SDK para .NET

Liste oportunidades.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// PDX-License-Identifier: Apache-2.0

using System;
using Newtonsoft.Json;
using Amazon;
using Amazon.Runtime;
using Amazon.PartnerCentralSelling;
using Amazon.PartnerCentralSelling.Model;

namespace AWSExample
{
```

```

class Program
{
    static readonly string catalogToUse = "Sandbox";
    static async Task Main(string[] args)
    {
        // Initialize credentials from .aws/credentials file
        var credentials = new
Amazon.Runtime.CredentialManagement.SharedCredentialsFile();
        if (credentials.TryGetProfile("default", out var profile))
        {
            AWSCredentials awsCredentials =
profile.GetAWSCredentials(credentials);

            //var config = new AmazonPartnerCentralSellingConfig()
            //{
            //    ServiceURL = "https://partnercentral-selling.us-
east-1.api.aws",
            //};
            //var client = new
AmazonPartnerCentralSellingClient(awsCredentials, config);
            var client = new
AmazonPartnerCentralSellingClient(awsCredentials);
            var request = new ListOpportunitiesRequest
            {
                Catalog = catalogToUse,
                MaxResults = 2
            };

            try {
                var response = await client.ListOpportunitiesAsync(request);
                Console.WriteLine(response.HttpStatusCode);
                foreach (var opportunity in response.OpportunitySummaries)
                {
                    Console.WriteLine("Opportunity id: " + opportunity.Id);
                }
                string formattedJson =
JsonConvert.SerializeObject(response.OpportunitySummaries, Formatting.Indented);
                Console.WriteLine(formattedJson);
            } catch (ValidationException ex) {
                Console.WriteLine("Validation error: " + ex.Message);
            } catch (AmazonPartnerCentralSellingException e) {
                Console.WriteLine("Failed:");
                Console.WriteLine(e.RequestId);
                Console.WriteLine(e.ErrorCode);
            }
        }
    }
}

```

```
        Console.WriteLine(e.Message);
    }
}
else
{
    Console.WriteLine("Profile not found.");
}
}
}
}
```

- Para obter detalhes da API, consulte [ListOpportunities](#) na Referência AWS SDK para .NET da API.

Go

SDK para Go V2

Liste oportunidades.

```
package main

import (
    "context"
    "encoding/json"
    "fmt"
    "log"

    "github.com/aws/aws-sdk-go-v2/aws"
    "github.com/aws/aws-sdk-go-v2/config"
    "github.com/aws/aws-sdk-go-v2/service/partnercentral-selling"
)

func main() {
    config, err := config.LoadDefaultConfig(context.TODO())

    if err != nil {
        log.Fatal(err)
    }

    config.Region = "us-east-1"
```

```

client := partnercentralselling.NewFromConfig(config)

output, err := client.ListOpportunities(context.TODO(),
&partnercentralselling.ListOpportunitiesInput{
    MaxResults: aws.Int32(2),
    Catalog:    aws.String("AWS"),
})

if err != nil {
    log.Fatal(err)
}

jsonOutput, err := json.MarshalIndent(output, "", "    ")
fmt.Println(string(jsonOutput))
}

```

- Para obter detalhes da API, consulte [ListOpportunities](#) a Referência AWS SDK para Go da API.

Java

SDK para Java 2.x

Liste oportunidades.

```

package org.example;

import java.util.ArrayList;
import java.util.List;

import org.example.utils.ReferenceCodesUtils;
import static org.example.utils.Constants.*;

import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.regions.Region;
import
    software.amazon.awssdk.services.partnercentralselling.PartnerCentralSellingClient;
import
    software.amazon.awssdk.services.partnercentralselling.model.ListOpportunitiesRequest;

```

```
import
    software.amazon.awssdk.services.partnercentralselling.model.ListOpportunitiesResponse;
import
    software.amazon.awssdk.services.partnercentralselling.model.OpportunitySummary;

/*
 * Purpose
 * PC-API-18 Getting list of Opportunities
 */

public class ListOpportunitiesPaging {

    static PartnerCentralSellingClient client =
        PartnerCentralSellingClient.builder()
            .region(Region.US_EAST_1)
            .credentialsProvider(DefaultCredentialsProvider.create())
            .httpClient(ApacheHttpClient.builder().build())
            .build();

    public static void main(String[] args) {
        List<OpportunitySummary> opportunitySummaries = getResponse();
        ReferenceCodesUtils.formatOutput(opportunitySummaries);
    }

    private static List<OpportunitySummary> getResponse() {
        List<OpportunitySummary> opportunitySummaries = new
        ArrayList<OpportunitySummary>();

        ListOpportunitiesRequest listOpportunityRequest =
        ListOpportunitiesRequest.builder()
            .catalog(CATALOG_T0_USE)
            .maxResults(5)
            .build();

        ListOpportunitiesResponse response =
        client.listOpportunities(listOpportunityRequest);

        opportunitySummaries.addAll(response.opportunitySummaries());

        while (response.nextToken() != null && response.nextToken().length() > 0) {
            listOpportunityRequest = ListOpportunitiesRequest.builder()
                .catalog(CATALOG_T0_USE)
                .maxResults(5)
                .nextToken(response.nextToken())
```

```

        .build();
        response = client.listOpportunities(listOpportunityRequest);
        opportunitySummaries.addAll(response.opportunitySummaries());
    }

    client.close();

    return opportunitySummaries;
}
}

```

- Para obter detalhes da API, consulte [ListOpportunities](#) na Referência AWS SDK for Java 2.x da API.

Python

SDK para Python (Boto3)

Liste oportunidades.

```

#!/usr/bin/env python

"""
Purpose
PC-API -18 Getting list of Opportunities
"""

import json
import logging
import boto3
import utils.helpers as helper

from utils.constants import CATALOG_TO_USE

serviceName = "partnercentral-selling"

partner_central_client = boto3.client(
    service_name=serviceName,
    region_name='us-east-1'
)

def get_list_of_opportunities():

```

```

    opportunity_list = []

    list_opportunities_request = {
        "Catalog": CATALOG_TO_USE,
        "MaxResults": 20
    }
    try:
        # Perform an API call
        response =
partner_central_client.list_opportunities(**list_opportunities_request)
        opportunity_list.extend(response["OpportunitySummaries"])

        while "NextToken" in response and response["NextToken"] is not None:
            list_opportunities_request["NextToken"] = response["NextToken"]
            response =
partner_central_client.list_opportunities(**list_opportunities_request)
            opportunity_list.extend(response["OpportunitySummaries"])

        return opportunity_list

    except Exception as err:
        # Catch all client exceptions
        print(json.dumps(err.response))

def usage_demo():
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Getting list of Opportunities.")
    print("-" * 88)

    helper.pretty_print_datetime(get_list_of_opportunities())

if __name__ == "__main__":
    usage_demo()

```

- Para obter detalhes da API, consulte a [ListOpportunities](#) Referência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Utilizar AWS API do Partner Central com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **ListSolutions** com um AWS SDK

Os exemplos de código a seguir mostram como usar o `ListSolutions`.

Java

SDK para Java 2.x

Recupera uma lista de soluções de parceiros que o parceiro registrou na Central de Parceiros.

```
package org.example;

import java.util.ArrayList;
import java.util.List;

import static org.example.utils.Constants.*;
import org.example.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.regions.Region;
import
    software.amazon.awssdk.services.partnercentralselling.PartnerCentralSellingClient;
import
    software.amazon.awssdk.services.partnercentralselling.model.ListSolutionsRequest;
import
    software.amazon.awssdk.services.partnercentralselling.model.ListSolutionsResponse;
import software.amazon.awssdk.services.partnercentralselling.model.SolutionBase;

/*
 * Purpose
 * PC-API-10 Getting list of solutions
 */

public class ListSolutions {

    static PartnerCentralSellingClient client =
        PartnerCentralSellingClient.builder()
            .region(Region.US_EAST_1)
```

```

        .credentialsProvider(DefaultCredentialsProvider.create())
        .httpClient(ApacheHttpClient.builder().build())
        .build();

    public static void main(String[] args) {
        List<SolutionBase> solutionSummaries = getResponse();
        ReferenceCodesUtils.formatOutput(solutionSummaries);
    }

    static List<SolutionBase> getResponse() {
        List<SolutionBase> solutionSummaries = new ArrayList<SolutionBase>();

        ListSolutionsRequest listSolutionsRequest = ListSolutionsRequest.builder()
            .catalog(CATALOG_T0_USE)
            .maxResults(5)
            .build();

        ListSolutionsResponse response = client.listSolutions(listSolutionsRequest);

        solutionSummaries.addAll(response.solutionSummaries());

        return solutionSummaries;
    }
}

```

- Para obter detalhes da API, consulte [ListSolutions](#) na Referência AWS SDK for Java 2.x da API.

Python

SDK para Python (Boto3)

Recupera uma lista de soluções de parceiros que o parceiro registrou na Central de Parceiros.

```

#!/usr/bin/env python

"""
Purpose
PC-API-10 Getting list of solutions
"""
import logging

```

```
import boto3
import utils.helpers as helper
from botocore.client import ClientError

from utils.constants import CATALOG_TO_USE

serviceName = "partnercentral-selling"

partner_central_client = boto3.client(
    service_name=serviceName,
    region_name='us-east-1'
)

def get_list_of_solutions():
    list_solutions_request = {
        "Catalog": CATALOG_TO_USE,
        "MaxResults": 20
    }
    try:
        # Perform an API call
        response =
partner_central_client.list_solutions(**list_solutions_request)
        return response

    except ClientError as err:
        # Catch all client exceptions
        print(err.response)

def usage_demo():
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Getting list of solutions.")
    print("-" * 88)

    helper.pretty_print_datetime(get_list_of_solutions())

if __name__ == "__main__":
    usage_demo()
```

- Para obter detalhes da API, consulte a [ListSolutions](#)Referência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Utilizar AWS API do Partner Central com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use `RejectEngagementInvitation` com um AWS SDK

Os exemplos de código a seguir mostram como usar o `RejectEngagementInvitation`.

Java

SDK para Java 2.x

Rejeita e `EngagementInvitation` isso é AWS compartilhado.

```
package org.example;

import static org.example.utils.Constants.*;

import org.example.utils.Constants;
import org.example.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.regions.Region;
import
    software.amazon.awssdk.services.partnercentralselling.PartnerCentralSellingClient;
import
    software.amazon.awssdk.services.partnercentralselling.model.RejectEngagementInvitationRe
import
    software.amazon.awssdk.services.partnercentralselling.model.RejectEngagementInvitationRe

/*
 * Purpose
 * PC-API-05 AWS Originated(A0) rejection
 */

public class RejectEngagementInvitation {

    static PartnerCentralSellingClient client =
        PartnerCentralSellingClient.builder()
            .region(Region.US_EAST_1)
            .credentialsProvider(DefaultCredentialsProvider.create())
            .httpClient(ApacheHttpClient.builder().build())
```

```

        .build();

    public static void main(String[] args) {

        String opportunityId = args.length > 0 ? args[0] : OPPORTUNITY_ID;

        RejectEngagementInvitationResponse response = getResponse(opportunityId);

        ReferenceCodesUtils.formatOutput(response);
    }

    static RejectEngagementInvitationResponse getResponse(String invitationId) {

        RejectEngagementInvitationRequest rejectOpportunityRequest =
        RejectEngagementInvitationRequest.builder()
            .catalog(Constants.CATALOG_TO_USE)
            .identifier(invitationId)
            .rejectionReason("Unable to support")
            .build();

        RejectEngagementInvitationResponse response =
        client.rejectEngagementInvitation(rejectOpportunityRequest);

        return response;
    }
}

```

- Para obter detalhes da API, consulte [RejectEngagementInvitation](#) na Referência AWS SDK for Java 2.x da API.

Python

SDK para Python (Boto3)

Rejeita e EngagementInvitation isso éAWS compartilhado.

```

#!/usr/bin/env python

"""
Purpose

```

PC-API-05 AWS Originated AO rejection - RejectOpportunityEngagementInvitation - Rejects a engagement invitation.

This action indicates that the partner does not wish to participate in the engagement and provides a reason for the rejection.

Upon rejection, a OpportunityEngagementInvitationRejected event is triggered. Subsequently, the invitation will no longer be available for the partner to act on.

"""

```
import json
import logging
import boto3
import utils.helpers as helper
```

```
from utils.constants import CATALOG_TO_USE
```

```
serviceName = "partnercentral-selling"
```

```
partner_central_client = boto3.client(
    service_name=serviceName,
    region_name='us-east-1'
)
```

```
def reject_opportunity_engagement_invitation(identifier, reject_reason):
```

```
    reject_opportunity_engagement_invitation_request = {
```

```
        "Catalog": CATALOG_TO_USE,
```

```
        "Identifier": identifier,
```

```
        "RejectionReason": reject_reason
```

```
    }
```

```
    try:
```

```
        # Perform an API call
```

```
        response =
```

```
partner_central_client.reject_engagement_invitation(**reject_opportunity_engagement_invi
```

```
        return response
```

```
    except Exception as err:
```

```
        # Catch all client exceptions
```

```
        print(json.dumps(err.response))
```

```
def usage_demo():
```

```
    identifier = "arn:aws:partnercentral:us-east-1::catalog/Sandbox/engagement-
invitation/engi-0000002isviga"
```

```
    reject_reason = "Customer problem unclear"
```

```
logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

print("-" * 88)
print("Given the ARN identifier and reject reason, reject the Opportunity
Engagement Invitation.")
print("-" * 88)

helper.pretty_print_datetime(reject_opportunity_engagement_invitation(identifier,
reject_reason))

if __name__ == "__main__":
    usage_demo()
```

- Para obter detalhes da API, consulte a [RejectEngagementInvitation](#) Referência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Utilizar AWS API do Partner Central com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **StartEngagementByAcceptingInvitationTask** com um AWS SDK

Os exemplos de código a seguir mostram como usar o **StartEngagementByAcceptingInvitationTask**.

Java

SDK para Java 2.x

Inicia o noivado aceitando um EngagementInvitation.

```
package org.example;

import static org.example.utils.Constants.*;

import org.example.utils.Constants;
import org.example.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
```

```

import software.amazon.awssdk.regions.Region;
import
    software.amazon.awssdk.services.partnercentralselling.PartnerCentralSellingClient;
import
    software.amazon.awssdk.services.partnercentralselling.model.StartEngagementByAcceptingIn
import
    software.amazon.awssdk.services.partnercentralselling.model.StartEngagementByAcceptingIn
import
    software.amazon.awssdk.services.partnercentralselling.model.GetEngagementInvitationReque
import
    software.amazon.awssdk.services.partnercentralselling.model.GetEngagementInvitationRespo
import
    software.amazon.awssdk.services.partnercentralselling.model.InvitationStatus;

/*
Purpose
PC-API-04: Start Engagement By Accepting InvitationTask for AWS Originated(A0)
    opportunity
*/

public class StartEngagementByAcceptingInvitationTask {

    static PartnerCentralSellingClient client =
        PartnerCentralSellingClient.builder()
            .region(Region.US_EAST_1)
            .credentialsProvider(DefaultCredentialsProvider.create())
            .httpClient(ApacheHttpClient.builder().build())
            .build();

    static String clientToken = "test-a30d161";

    public static void main(String[] args) {

        String opportunityId = args.length > 0 ? args[0] : OPPORTUNITY_ID;

        StartEngagementByAcceptingInvitationTaskResponse response =
            getResponse(opportunityId);

        if ( response == null) {
            System.out.println("Opportunity is not AWS Originated.");
        } else {
            ReferenceCodesUtils.formatOutput(response);
        }
    }
}

```

```
private static GetEngagementInvitationResponse getInvitation(String
invitationId) {

    GetEngagementInvitationRequest getRequest =
    GetEngagementInvitationRequest.builder()
        .catalog(Constants.CATALOG_TO_USE)
        .identifier(invitationId)
        .build();

    GetEngagementInvitationResponse response =
    client.getEngagementInvitation(getRequest);

    return response;
}

static StartEngagementByAcceptingInvitationTaskResponse getResponse(String
invitationId) {

    if ( getInvitation(invitationId).status().equals(InvitationStatus.PENDING)) {
        StartEngagementByAcceptingInvitationTaskRequest acceptOpportunityRequest =
        StartEngagementByAcceptingInvitationTaskRequest.builder()
            .catalog(Constants.CATALOG_TO_USE)
            .identifier(invitationId)
            .clientToken(clientToken)
            .build();

        StartEngagementByAcceptingInvitationTaskResponse response =
        client.startEngagementByAcceptingInvitationTask(acceptOpportunityRequest);
        return response;
    }
    return null;
}
}
```

- Para obter detalhes da API, consulte [StartEngagementByAcceptingInvitationTask](#) na Referência AWS SDK for Java 2.x da API.

Python

SDK para Python (Boto3)

Inicia o noivado aceitando um EngagementInvitation.

```
#!/usr/bin/env python

"""
Purpose
PC-API -11 Associating a product
PC-API -12 Associating a solution
PC-API -13 Associating an offer
"""

import logging
import boto3
import utils.helpers as helper
from botocore.client import ClientError

from utils.constants import CATALOG_TO_USE

serviceName = "partnercentral-selling"

partner_central_client = boto3.client(
    service_name=serviceName,
    region_name='us-east-1'
)

def get_opportunity(identifier):
    get_opportunity_request = {
        "Identifier": identifier,
        "Catalog": CATALOG_TO_USE
    }
    try:
        # Perform an API call
        response =
partner_central_client.get_engagement_invitation(**get_opportunity_request)
        return response

    except ClientError as err:
        # Catch all client exceptions
        print(err.response)
```

```

def start_engagement_by_accepting_invitation_task(identifier):

    response = get_opportunity(identifier)

    if ( response['Status'] == 'PENDING') :
        accept_opportunity_engagement_invitation_request = {
            "Catalog": CATALOG_TO_USE,
            "Identifier" : identifier,
            "ClientToken": "test-123456"
        }
        try:
            # Perform an API call
            response =
partner_central_client.start_engagement_by_accepting_invitation_task(**accept_opportunity_engagement_invitation_request)
            return response

        except ClientError as err:
            # Catch all client exceptions
            print(err.response)
            return None
    else:
        return None

def usage_demo():
    identifier = "arn:aws:partnercentral:us-east-1::catalog/Sandbox/engagement-
invitation/engi-0000002isusga"
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Get updated Opportunity.")
    print("-" * 88)

    helper.pretty_print_datetime(start_engagement_by_accepting_invitation_task(identifier))

if __name__ == "__main__":
    usage_demo()

```

- Para obter detalhes da API, consulte a [StartEngagementByAcceptingInvitationTask](#) Referência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Utilizar AWS API do Partner Central com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **StartEngagementFromOpportunityTask** com um AWS SDK

Os exemplos de código a seguir mostram como usar o **StartEngagementFromOpportunityTask**.

Java

SDK para Java 2.x

Inicia o processo de engajamento com base em uma oportunidade existente aceitando o convite de engajamento e criando uma oportunidade correspondente no sistema do parceiro.

```
package org.example;

import static org.example.utils.Constants.*;

import org.example.utils.Constants;
import org.example.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.regions.Region;
import
    software.amazon.awssdk.services.partnercentralselling.PartnerCentralSellingClient;
import software.amazon.awssdk.services.partnercentralselling.model.AwsSubmission;
import
    software.amazon.awssdk.services.partnercentralselling.model.SalesInvolvementType;
import
    software.amazon.awssdk.services.partnercentralselling.model.StartEngagementFromOpportunityTask;
import
    software.amazon.awssdk.services.partnercentralselling.model.StartEngagementFromOpportunityTaskRequest;
import software.amazon.awssdk.services.partnercentralselling.model.Visibility;

/*
 * Purpose
 * PC-API-01 Partner Originated (PO) opp submission(Start Engagement From
 * Opportunity Task for A0 Originated Opportunity)
 */
```

```

public class StartEngagementFromOpportunityTask {

    static PartnerCentralSellingClient client =
        PartnerCentralSellingClient.builder()
            .region(Region.US_EAST_1)
            .credentialsProvider(DefaultCredentialsProvider.create())
            .httpClient(ApacheHttpClient.builder().build())
            .build();

    public static void main(String[] args) {

        String opportunityId = args.length > 0 ? args[0] : OPPORTUNITY_ID;

        StartEngagementFromOpportunityTaskResponse response =
            getResponse(opportunityId);

        ReferenceCodesUtils.formatOutput(response);
    }

    static StartEngagementFromOpportunityTaskResponse getResponse(String
        opportunityId) {

        StartEngagementFromOpportunityTaskRequest submitOpportunityRequest =
            StartEngagementFromOpportunityTaskRequest.builder()
                .catalog(Constants.CATALOG_T0_USE)
                .identifier(opportunityId)
                .clientToken("test-annjqwesdsd99")

                .awsSubmission(AwsSubmission.builder().involvementType(SalesInvolvementType.CO_SELL).vis
                    .build());

        StartEngagementFromOpportunityTaskResponse response =
            client.startEngagementFromOpportunityTask(submitOpportunityRequest);

        return response;
    }
}

```

- Para obter detalhes da API, consulte [StartEngagementFromOpportunityTask](#) a Referência AWS SDK for Java 2.x da API.

Python

SDK para Python (Boto3)

Inicia o processo de engajamento com base em uma oportunidade existente aceitando o convite de engajamento e criando uma oportunidade correspondente no sistema do parceiro.

```
#!/usr/bin/env python

"""
Purpose
PC-API -11 Associating a product
PC-API -12 Associating a solution
PC-API -13 Associating an offer
"""

import logging
import boto3
import utils.helpers as helper
from botocore.client import ClientError

from utils.constants import CATALOG_TO_USE

serviceName = "partnercentral-selling"

partner_central_client = boto3.client(
    service_name=serviceName,
    region_name='us-east-1'
)

def start_engagement_from_opportunity_task(identifier):

    start_engagement_from_opportunity_task_request = {
        "AwsSubmission": {
            "InvolvementType": "Co-Sell",
            "Visibility": "Full"
        },
        "Catalog": CATALOG_TO_USE,
        "Identifier" : identifier,
        "ClientToken": "test-annjqwesdsd99"
    }
    try:
        # Perform an API call
```

```

        response =
partner_central_client.start_engagement_from_opportunity_task(**start_engagement_from_op
        return response

    except ClientError as err:
        # Catch all client exceptions
        print(err.response)
        return None

def usage_demo():
    identifier = "05465588"

    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Start Engagement from Opportunity Task.")
    print("-" * 88)

    helper.pretty_print_datetime(start_engagement_from_opportunity_task(identifier))

if __name__ == "__main__":
    usage_demo()

```

- Para obter detalhes da API, consulte a [StartEngagementFromOpportunityTask](#) Referência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Utilizar AWS API do Partner Central com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **UpdateOpportunity** com um AWS SDK

Os exemplos de código a seguir mostram como usar o UpdateOpportunity.

Java

SDK para Java 2.x

Atualize uma oportunidade.

```
package org.example;

import java.time.Instant;
import java.util.ArrayList;
import java.util.List;

import static org.example.utils.Constants.*;

import org.example.entity.Root;
import org.example.utils.Constants;
import org.example.utils.ReferenceCodesUtils;
import org.example.utils.StringSerializer;

import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.regions.Region;
import
    software.amazon.awssdk.services.partnercentralselling.PartnerCentralSellingClient;
import software.amazon.awssdk.services.partnercentralselling.model.Account;
import software.amazon.awssdk.services.partnercentralselling.model.Address;
import software.amazon.awssdk.services.partnercentralselling.model.Contact;
import software.amazon.awssdk.services.partnercentralselling.model.Customer;
import
    software.amazon.awssdk.services.partnercentralselling.model.ExpectedCustomerSpend;
import
    software.amazon.awssdk.services.partnercentralselling.model.GetOpportunityRequest;
import
    software.amazon.awssdk.services.partnercentralselling.model.GetOpportunityResponse;
import software.amazon.awssdk.services.partnercentralselling.model.Lifecycle;
import software.amazon.awssdk.services.partnercentralselling.model.Marketing;
import
    software.amazon.awssdk.services.partnercentralselling.model.NextStepsHistory;
import software.amazon.awssdk.services.partnercentralselling.model.Project;
import software.amazon.awssdk.services.partnercentralselling.model.ReviewStatus;
import
    software.amazon.awssdk.services.partnercentralselling.model.UpdateOpportunityRequest;
import
    software.amazon.awssdk.services.partnercentralselling.model.UpdateOpportunityResponse;

import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
import com.google.gson.ToNumberPolicy;
```

```
/*
 * Purpose
 * PC-API-02/06 Update opportunity when Lifecycle.ReviewStatus is not Submitted
 * or In-Review
 */

public class UpdateOpportunity {

    static final Gson GSON = new GsonBuilder()
        .setObjectToNumberStrategy(ToNumberPolicy.LAZILY_PARSED_NUMBER)
        .registerTypeAdapter(String.class, new StringSerializer())
        .create();

    static PartnerCentralSellingClient client =
        PartnerCentralSellingClient.builder()
            .region(Region.US_EAST_1)
            .credentialsProvider(DefaultCredentialsProvider.create())
            .httpClient(ApacheHttpClient.builder().build())
            .build();

    static String OPPORTUNITY_ORIGIN = ORIGIN_PARTNER_ORIGINATED;

    public static void main(String[] args) {

        String inputFile = "updateOpportunity.json";

        if (args.length > 0)
            inputFile = args[0];

        UpdateOpportunityResponse response = updateOpportunity(inputFile);

        client.close();
    }

    public static GetOpportunityResponse getResponse(String opportunityId) {

        GetOpportunityRequest getOpportunityRequest =
            GetOpportunityRequest.builder()
                .catalog(Constants.CATALOG_TO_USE)
                .identifier(opportunityId)
                .build();

        GetOpportunityResponse response =
            client.getOpportunity(getOpportunityRequest);
    }
}
```

```

        System.out.println(opportunityId + ":" + response);
        return response;
    }

    public static UpdateOpportunityResponse updateOpportunity(String inputFile) {

        String inputString = ReferenceCodesUtils.readInputFileToString(inputFile);

        Root root = GSON.fromJson(inputString, Root.class);
        GetOpportunityResponse response = getResponse(root.identifier);

        if (response != null
            && response.lifeCycle() != null
            && response.lifeCycle().reviewStatus() != null
            && response.lifeCycle().reviewStatus() != ReviewStatus.SUBMITTED
            && response.lifeCycle().reviewStatus() != ReviewStatus.IN_REVIEW) {

            List<NextStepsHistory> nextStepsHistories = new ArrayList<NextStepsHistory>();
            if ( root.lifeCycle != null && root.lifeCycle.nextStepsHistories != null) {
                for (org.example.entity.NextStepsHistory nextStepsHistoryJson :
                    root.lifeCycle.nextStepsHistories) {
                    NextStepsHistory nextStepsHistory = NextStepsHistory.builder()
                        .time(Instant.parse(nextStepsHistoryJson.time))
                        .value(nextStepsHistoryJson.value)
                        .build();
                    nextStepsHistories.add(nextStepsHistory);
                }
            }

            LifeCycle lifeCycle = null;
            if ( root.lifeCycle != null ) {
                lifeCycle = LifeCycle.builder()
                    .closedLostReason(root.lifeCycle.closedLostReason)
                    .nextSteps(root.lifeCycle.nextSteps)
                    .nextStepsHistory(nextStepsHistories)
                    .reviewComments(root.lifeCycle.reviewComments)
                    .reviewStatus(root.lifeCycle.reviewStatus)
                    .reviewStatusReason(root.lifeCycle.reviewStatusReason)
                    .stage(root.lifeCycle.stage)
                    .targetCloseDate(root.lifeCycle.targetCloseDate)
                    .build();
            }

            Marketing marketing = null;

```

```

if ( root.marketing != null ) {
    marketing = Marketing.builder()
        .awsFundingUsed(root.marketing.awsFundingUsed)
        .campaignName(root.marketing.campaignName)
        .channels(root.marketing.channels)
        .source(root.marketing.source)
        .useCases(root.marketing.useCases)
        .build();
}

Address address = null;
if (root.customer != null && root.customer.account != null &&
root.customer.account.address != null) {
    address =
Address.builder().postalCode(root.customer.account.address.postalCode)
    .stateOrRegion(root.customer.account.address.stateOrRegion)
    .countryCode(root.customer.account.address.countryCode).build();
}

Account account = null;
if (root.customer != null && root.customer.account != null) {
    account = Account.builder().address(address).duns(root.customer.account.duns)

.industry(root.customer.account.industry).companyName(root.customer.account.companyName)
    .websiteUrl(root.customer.account.websiteUrl).build();
}

List<Contact> contacts = new ArrayList<Contact>();
if ( root.customer != null && root.customer.contacts != null) {
    for (org.example.entity.Contact jsonContact : root.customer.contacts) {
        Contact contact = Contact.builder()
            .email(jsonContact.email)
            .firstName(jsonContact.firstName)
            .lastName(jsonContact.lastName)
            .phone(jsonContact.phone)
            .businessTitle(jsonContact.businessTitle)
            .build();
        contacts.add(contact);
    }
}

Customer customer =
Customer.builder().account(account).contacts(contacts).build();

```

```

    List<ExpectedCustomerSpend> expectedCustomerSpends = new
    ArrayList<ExpectedCustomerSpend>();
    if ( root.project != null && root.project.expectedCustomerSpend != null) {
        for (org.example.entity.ExpectedCustomerSpend expectedCustomerSpendJson :
root.project.expectedCustomerSpend) {
            ExpectedCustomerSpend expectedCustomerSpend = null;
            expectedCustomerSpend = ExpectedCustomerSpend.builder()
                .amount(expectedCustomerSpendJson.amount)
                .currencyCode(expectedCustomerSpendJson.currencyCode)
                .frequency(expectedCustomerSpendJson.frequency)
                .targetCompany(expectedCustomerSpendJson.targetCompany)
                .build();
            expectedCustomerSpends.add(expectedCustomerSpend);
        }
    }

    Project project = null;
    if (root.project != null) {
        project = Project.builder().title(root.project.title)
            .customerBusinessProblem(root.project.customerBusinessProblem)

.customerUseCase(root.project.customerUseCase).deliveryModels(root.project.deliveryModel
            .expectedCustomerSpend(expectedCustomerSpends)

.salesActivities(root.project.salesActivities).competitorName(root.project.competitorName
            .otherSolutionDescription(root.project.otherSolutionDescription).build();
    }

    // Building the Actual CreateOpportunity Request
    UpdateOpportunityRequest updateOpportunityRequest =
    UpdateOpportunityRequest.builder().catalog(root.catalog)

.identifier(root.identifier).lastModifiedDate(Instant.parse(root.lastModifiedDate))

.primaryNeedsFromAwsWithStrings(root.primaryNeedsFromAws).opportunityType(root.opportuni
        .lifeCycle(lifeCycle)
        .customer(customer)
        .project(project)
        .partnerOpportunityIdentifier(root.partnerOpportunityIdentifier)
        .marketing(marketing)
        .nationalSecurity(root.nationalSecurity)
        .opportunityType(root.opportunityType)
        .build();

```

```

        UpdateOpportunityResponse updateResponse =
client.updateOpportunity(updateOpportunityRequest);
        System.out.println("Successfully updated opportunity: " + updateResponse);

        return updateResponse;
    } else {
        System.out.println("Opportunity cannot be updated.");
        return null;
    }
}
}
}

```

- Para obter detalhes da API, consulte [UpdateOpportunity](#) a Referência AWS SDK for Java 2.x da API.

Python

SDK para Python (Boto3)

Atualize uma oportunidade.

```

#!/usr/bin/env python

"""
Purpose
PC-API-2 Updating Partner Originated Opportunity
"""

import logging
import boto3
import sys
import os
sys.path.append(os.path.dirname(os.path.dirname(os.path.abspath(__file__))))
import utils.helpers as helper
from botocore.client import ClientError
import utils.stringify_details as sd
from utils.constants import CATALOG_TO_USE

serviceName = "partnercentral-selling"

partner_central_client = boto3.client(

```

```
        service_name=serviceName,
        region_name='us-east-1'
    )

def get_opportunity(identifier):
    get_opportunity_request = {
        "Identifier": identifier,
        "Catalog": CATALOG_TO_USE
    }
    try:
        # Perform an API call
        response =
partner_central_client.get_opportunity(**get_opportunity_request)
        return response

    except ClientError as err:
        # Catch all client exceptions
        print(err.response)

def update_opportunity():
    update_opportunity_request_orig = sd.stringify_json("src/update_opportunity/
update_opportunity_technical_validation.json")
    update_opportunity_request =
helper.remove_nulls(update_opportunity_request_orig)

    try:
        # Perform an API call
        response =
partner_central_client.update_opportunity(**update_opportunity_request)
        return response

    except ClientError as err:
        # Catch all client exceptions
        print(err.response)

def update_opportunity_if_eligible(identifier):
    response = get_opportunity(identifier)
    if response is not None:
        return update_opportunity()
    else:
        print("Failed to retrieve opportunity details")

def usage_demo():
    identifier = "05465588"
```

```
logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

print("-" * 88)
print("Updating opportunity.")
print("-" * 88)

helper.pretty_print_datetime(update_opportunity_if_eligible(identifier))

if __name__ == "__main__":
    usage_demo()
```

- Para obter detalhes da API, consulte a [UpdateOpportunity](#) Referência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Utilizar AWS API do Partner Central com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Cenários para o uso do Partner Central AWS SDKs

Os exemplos de código a seguir mostram como implementar cenários comuns no Partner Central com AWS SDKs. Esses cenários mostram como realizar tarefas específicas chamando várias funções na Central de Parceiros ou combinadas com outros Serviços da AWS. Cada cenário inclui um link para o código-fonte completo, onde podem ser encontradas instruções sobre como configurar e executar o código.

Os cenários têm como alvo um nível intermediário de experiência para ajudar você a compreender ações de serviço em contexto.

Exemplos

- [Atualizar a entidade associada de uma oportunidade](#)

Atualizar a entidade associada de uma oportunidade

Os exemplos de código a seguir mostram como:

- Desassociar uma entidade antiga.
- Associar uma entidade nova.

Java

SDK para Java 2.x

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no repositório [Cenários](#).

Atualizar a entidade associada de uma oportunidade

```
package org.example;

import static org.example.utils.Constants.*;

import org.example.utils.Constants;
import org.example.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.regions.Region;
import
    software.amazon.awssdk.services.partnercentralselling.PartnerCentralSellingClient;
import
    software.amazon.awssdk.services.partnercentralselling.model.AssociateOpportunityRequest;
import
    software.amazon.awssdk.services.partnercentralselling.model.AssociateOpportunityResponse;
import
    software.amazon.awssdk.services.partnercentralselling.model.DisassociateOpportunityRequest;
import
    software.amazon.awssdk.services.partnercentralselling.model.DisassociateOpportunityResponse;

/*
Purpose
PC-API -17 Replacing a solution
*/

public class ReplaceSolution {

    static PartnerCentralSellingClient client =
        PartnerCentralSellingClient.builder()
            .region(Region.US_EAST_1)
```

```

        .credentialsProvider(DefaultCredentialsProvider.create())
        .httpClient(ApacheHttpClient.builder().build())
        .build();

public static void main(String[] args) {

    String opportunityId = args.length > 0 ? args[0] : OPPORTUNITY_ID;

    String entityType = "Solutions";
    String originalEntityIdentifier = "S-00000000";
    String newEntityIdentifier = "S-00111111";

    disassociateOppornitityResponse(opportunityId, entityType,
originalEntityIdentifier );
    AssociateOppportunityResponse associateOppportunityResponse =
associateOppportunityResponse(opportunityId, entityType, newEntityIdentifier );

    ReferenceCodesUtils.formatOutput(associateOppportunityResponse);
}

private static AssociateOppportunityResponse associateOppportunityResponse(String
opportunityId, String entityType, String entityIdentifier) {

    AssociateOppportunityRequest associateOppportunityRequest =
AssociateOppportunityRequest.builder()
        .catalog(Constants.CATALOG_TO_USE)
        .opportunityIdentifier(opportunityId)
        .relatedEntityType(entityType)
        .relatedEntityIdentifier(entityIdentifier)
        .build();

    AssociateOppportunityResponse response =
client.associateOppportunity(associateOppportunityRequest);

    return response;
}

private static DisassociateOppportunityResponse
disassociateOppornitityResponse(String opportunityId, String entityType, String
entityIdentifier) {
    PartnerCentralSellingClient client = PartnerCentralSellingClient.builder()
        .region(Region.US_EAST_1)
        .credentialsProvider(DefaultCredentialsProvider.create())
        .httpClient(ApacheHttpClient.builder().build())

```

```
        .build();

        DisassociateOpportunityRequest disassociateOpportunityRequest =
        DisassociateOpportunityRequest.builder()
            .catalog(Constants.CATALOG_TO_USE)
            .opportunityIdentifier(opportunityId)
            .relatedEntityType(entityType)
            .relatedEntityIdentifier(entityIdentifier)
            .build();

        DisassociateOpportunityResponse response =
        client.disassociateOpportunity(disassociateOpportunityRequest);

        return response;
    }
}
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK for Java 2.x.
 - [AssociateOpportunity](#)
 - [DisassociateOpportunity](#)

Python

SDK para Python (Boto3)

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [AWS Code Examples Repository](#).

Atualizar a entidade associada de uma oportunidade

```
#!/usr/bin/env python

"""
Purpose
PC-API -17 Replacing a solution
```

```

"""
import logging
import boto3
import utils.helpers as helper
from botocore.client import ClientError

from utils.constants import CATALOG_TO_USE

serviceName = "partnercentral-selling"

partner_central_client = boto3.client(
    service_name=serviceName,
    region_name='us-east-1'
)

def replace_solution(original_entity_identifier, new_entity_identifier,
    opportunityIdentifier):
    disassociate_opportunity_request = {
        "Catalog": CATALOG_TO_USE,
        "OpportunityIdentifier" : opportunityIdentifier,
        "RelatedEntityType" : "Solutions",
        "RelatedEntityIdentifier" : original_entity_identifier
    }

    associate_opportunity_request = {
        "Catalog": CATALOG_TO_USE,
        "OpportunityIdentifier" : opportunityIdentifier,
        "RelatedEntityType" : "Solutions",
        "RelatedEntityIdentifier" : new_entity_identifier
    }
    try:
        # Perform an API call
        response =
partner_central_client.disassociate_opportunity(**disassociate_opportunity_request)
        response =
partner_central_client.associate_opportunity(**associate_opportunity_request)
        return response

    except ClientError as err:
        # Catch all client exceptions
        print(err.response)

def usage_demo():
    original_entity_identifier = "S-0049999"

```

```
new_entity_identifier = "S-0050014"
opportunityIdentifier = "04397574"

logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

print("-" * 88)
print("Replacing a solution.")
print("-" * 88)

helper.pretty_print_datetime(replace_solution(original_entity_identifier,
new_entity_identifier, opportunityIdentifier))

if __name__ == "__main__":
    usage_demo()
```

- Consulte detalhes da API nos tópicos a seguir na Referência de API do AWS SDK para Python (Boto3).
 - [AssociateOpportunity](#)
 - [DisassociateOpportunity](#)

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulte [Utilizar AWS API do Partner Central com um AWS SDK](#). Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Notas da versão

Esta página contém um resumo das mudanças significativas na documentação da API do AWS Partner Central, começando com a atualização mais recente.

Histórico de revisões

Alteração	Descrição	Data
ARCO 6.0	O AWS Partner Central adicionou novas APIs de medição de receita de parceiros.	2026-06-30
ARCO 5.3	Atualizações do ACE Resonate: GetAwsOppportunitySummary resposta atualizada.	2026-06-16
ARCO 5.2	- Suporte monetário EUR: CreateOppportunity UpdateOppportunity , e CreateEngagementInvitation agora aceita, EUR além do código USD de moeda MRR. Quando a partição da AWS é aws-eusc (AWS European Sovereign Cloud), o código da moeda deve serEUR. - EUR em operações de leitura: GetOppportunity ListOpportunities , e GetAwsOppportunitySummary agora retorne EUR ao	2026-03-30

Alteração	Descrição	Data
	CurrencyCode campo para oportunidades na aws-eusc partição.	
ARCO 5.1	• Suporte monetário EUR: CreateOpportunity UpdateOpportunity ,, e CreateEngagementInvitation agora aceita, EUR além do código USD de moeda MRR. Quando a partição da AWS é aws-eusc (AWS European Sovereign Cloud), o código da moeda deve ser EUR. • EUR em operações de leitura: GetOpportunity ListOpportunities ,, e GetAwsOpportunitySummary agora retorne EUR ao CurrencyCode campo para oportunidades na aws-eusc partição.	2026-03-30

Alteração	Descrição	Data
ARCO 5	• API de conta introduzida: foi adicionada uma nova API de conta ao AWS Partner Central, permitindo que os parceiros gerenciem as informações da conta, o registro, os perfis, o gerenciamento de domínio e as conexões de parceiros. • Documentação de referência da API da conta: foi adicionada uma documentação abrangente de referência da API para a API da conta, incluindo registro de parceiros, gerenciamento de perfil, verificação de domínio e fluxos de trabalho de conexão de conta. • Registro da API da conta: foi adicionado suporte de registro e documentação para a API da conta para ajudar a auditar e monitorar o uso da API. • Notificações da API da conta: foi adicionado suporte à notificação de eventos para a API da conta, permitindo que os parceiros recebam atualizações em tempo real sobre as atividades relacionadas à	2025-11-30

Alteração	Descrição	Data
	conta, incluindo conexões da conta do parceiro. • Cotas da API da conta: foi adicionada a documentação de cotas de serviço para a API da conta. • Teste de sandbox para a API de contas: foram adicionados recursos de teste de sandbox para a API de contas, permitindo que os parceiros testem fluxos de trabalho de gerenciamento de contas em um ambiente que não seja de produção. • API de benefícios introduzida: foi adicionada uma nova API de benefícios ao AWS Partner Central, permitindo que os parceiros descubram, se inscrevam e gerenciem benefícios em todos os programas de AWS parceiros por meio de uma interface centralizada. • Documentação de referência da API de benefícios: foi adicionada uma documentação de referência abrangente da API de benefícios para a API de benefícios, alocações e processos de atendimento.	

Alteração	Descrição	Data
	• Registro da API de benefícios: foi adicionado o suporte de registro e documentação para a API de benefícios para ajudar a auditar e monitorar o uso da API. • Notificações da API de benefícios: foi adicionado o suporte à notificação de eventos para a API de benefícios, permitindo que os parceiros recebam atualizações em tempo real sobre solicitações e alocações de benefícios. • Cotas da API de benefícios: foi adicionada a documentação de cotas de serviço para a API de benefícios. • Teste de sandbox para API de benefícios: foram adicionados recursos de teste de sandbox para a API de benefícios, permitindo que os parceiros testem fluxos de trabalho de gerenciamento de benefícios em um ambiente não produtivo. • API de vendas atualizada: ações aprimoradas de engajamento da API de vendas para apoiar o fluxo	

Alteração	Descrição	Data
	de trabalho de gerenciamento de leads. • Atualizações da documentação da API de venda: documentação aprimorada da API de vendas com estrutura aprimorada e fluxos de trabalho atualizados para gerenciamento de leads. • Venda de atualizações de notificações da API: documentação aprimorada de notificação de eventos para eventos de engajamento e convite de engajamento. • Expansão da estrutura da API: estrutura de documentação expandida para oferecer suporte a quatro APIs distintas (conta, benefícios, vendas e canal) com seções dedicadas para as permissões, registros, notificações, cotas e testes de sandbox de cada API. • Documentação aprimorada de uso da API: foram adicionados guias de uso abrangentes para APIs de contas e benefícios com fluxos de trabalho detalhados e padrões de integração.	

Alteração	Descrição	Data
	• Documentação das regiões suportadas: foi adicionada documentação para AWS as regiões compatíveis com as APIs de contas e benefícios. • Novo filtro para <code>ListOpportunities : TargetCloseDate</code> filtro adicionado para permitir filtrar oportunidades de acordo com a data de fechamento esperada, uso <code>AfterTargetCloseDate</code> e <code>BeforeTargetCloseDate</code> campos em YYYY-MM-DD formato.	

Alteração	Descrição	Data
ARCO 4	• API de canal introduzida: foi adicionada uma nova API de canal ao AWS Partner Central, permitindo que os parceiros gerenciem relacionamentos entre canais e colaborações entre parceiros. • Documentação de referência da API do canal: foi adicionada uma documentação de referência abrangente da API do canal, incluindo todas as ações e modelos de dados disponíveis. • Permissões da API do canal: foram adicionadas políticas de permissão do IAM e documentação de controle de acesso específicas à API do canal. • Registro da API do canal: foi adicionado suporte de registro e documentação para a API do canal para ajudar a auditar e monitorar o uso da API. • Notificações da API do canal: foi adicionado suporte à notificação de eventos para a API do canal, permitindo que os parceiros recebam atualizações em tempo real sobre as	2025-11-19

Alteração	Descrição	Data
	atividades relacionadas ao canal. • Cotas da API do canal: foi adicionada a documentação de cotas de serviço para a API do canal. • Teste de sandbox para a API do canal: foram adicionados recursos de teste de sandbox para a API do canal, permitindo que os parceiros testem os fluxos de trabalho do canal em um ambiente não produtivo. • Reorganização da estrutura da API: documentação reorganizada para distinguir entre a API de venda e a API de canal, com seções dedicadas para as permissões, registros, notificações e cotas de cada API. • Documentação das regiões suportadas: foi adicionada documentação para AWS regiões compatíveis com as APIs de vendas e de canal.	

Alteração	Descrição	Data
ARCO 3	• Políticas de permissão de API atualizadas: as políticas foram atualizadas para refletir as novas ações introduzidas nesta versão. Certifique-se de que suas funções e permissões do IAM sejam ajustadas adequadamente. • Detalhes de uso do cabeçalho personalizado: foram adicionados detalhes sobre como usar o cabeçalho personalizado X-Amzn-User-Agent para ajudar a auditar e medir o uso da API. • Ações substituídas por ações assíncronas: <code>AcceptOpportunityEngagementInvitation</code> Substituído pela ação assíncrona <code>StartEngagementByAcceptingInvitationTask</code> e o formato da carga foi alterado em relação ao RC2. • Renomeação de entidades e ações: Alterado <code>OpportunityEngagementInvitation</code> para <code>EngagementInvitation</code>. Os	2024-10-17

Alteração	Descrição	Data
	atributos dessas ações também foram alterados para se alinharem à nova entidade. As seguintes ações foram substituídas: • <code>GetOpportunityEngagementInvitation</code> é agora <code>GetEngagementInvitation</code> . Vários atributos no convite foram alterados, adicionados ou removidos . • <code>ListOpportunityEngagementInvitations</code> é agora <code>ListEngagementInvitations</code> . • <code>RejectOpportunityEngagementInvitation</code> é agora <code>RejectEngagementInvitation</code> . • <code>AssignOpportunity</code> atualização de parâmetros: <code>AssignOpportunity</code> agora leva <code>Assignee</code> em vez de <code>AssigneeEmail</code> . Atualize suas implementações adequadamente. • <code>SubmitOpportunity</code> alteração de funcionalidade: agora	

Alteração	Descrição	Data
	<code>SubmitOpportunity</code> é executada pela ação assíncrona <code>StartEngagementFromOpportunityTask</code>. <code>SubmitOpportunity</code> agora aceita <code>InvolvementType</code> e <code>Visibility</code>. • Renomeação e expansão de atributos: <code>EstimatedAwsMonthlyRevenue</code> alterado para <code>ExpectedCustomerSpend</code>. <code>ExpectedCustomerSpend</code> agora inclui novos atributos, como <code>Frequency</code> e <code>TargetCompany</code>. • Resumo da nova ação para AWS oportunidade: a visibilidade da oportunidade agora é fornecida por meio de uma ação separada chamada <code>GetAwsOpportunitySummary</code>. • Atualizações do fluxo de trabalho para trabalhar com oportunidades: O fluxo de trabalho para trabalhar com suas oportunidades foi atualizado para usar as novas ações introduzidas nesta versão. • Atualizações de fluxo de trabalho para trabalhar com	

Alteração	Descrição	Data
	oportunidades: o fluxo de trabalho para trabalhar com oportunidades foi atualizado para usar <code>EngagementInvitations</code>, o que melhora as oportunidades de tratamento oferecidas pela AWS. • Atualizações do fluxo de trabalho para rastrear atualizações: O fluxo de trabalho para rastrear atualizações agora está trabalhando com atualizações de oportunidades, aumentando a clareza e a eficiência no monitoramento do status da oportunidade. • Novos filtros para listar ações: novos filtros estão disponíveis para <code>ListOpportunities</code> e <code>ListSolutions</code>. • <code>PoisListSolutions</code>, os novos filtros são <code>FilterCategory</code>, <code>FilterIdentifier</code>, <code>FilterStatus</code> e. • <code>PoisListOpportunities</code>, os novos filtros são <code>FilterCompanyName</code>, <code>FilterId</code>	

Alteração	Descrição	Data
	ntifier , FilterLas tModified Date FilterLif eCycleApp rovalStat us ,FilterLif eCycleStage ,, FilterOrigin e. • Remoção do AccessCon trol subobjeto: O AccessControl subobjeto no modelo de oportunidade foi removido. NationalS ecurity agora é um atributo de nível superior. • Atributos realocado s paraGetAwsOpp portunitySummary : Atributos como AWSOpportunityTeam insights (Engagemen tScore ,NextBestA ction), origem e agora InvolvementType estão disponíveis por meio de, GetAwsOpp portunitySummary em vez deGetOpportunity . • Introdução do InvolvementType campo: introduzi u o InvolvementType campo para diferenciar entre oportunidades de	

Alteração	Descrição	Data
	venda conjunta e oportunidades somente de visibilidade. • Atualizações do nome do evento: os eventos foram atualizados para refletir as mudanças. Por exemplo, o convite de engajamento de oportunidade criado agora é um convite de engajamento criado. • Atualizações da carga útil de resposta: todas as cargas de resposta agora incluem um ARN and/or de ID. • Solicitar atualizações de carga útil: todas as cargas de solicitação agora usam identificadores como entrada, que aceitam um ID ou ARN. • Atualização da entidade de engajamento: o atributo de título na entidade de engajamento foi renomeado <code>EngagementTitle</code>. • Remoção do prefixo do filtro: todos os filtros não incluem mais o prefixo do filtro, simplificando o mecanismo de filtragem. • <code>StartEngagementFromOpportunity</code> atualizaç	

Alteração	Descrição	Data
	ção da ação: o atributo <code>AwsInvolvement</code> foi alterado para <code>AwsSubmission</code> . • Atualização dos detalhes do projeto de convite: <code>Invitation.ProjectDetails.TargetCompletionDate</code> foi renomeado <code>Invitation.ProjectDetails.EstimatedCompletionDate</code> . • Atualização da estimativa de receita: <code>Invitation.PartnerRevenueEstimate</code> é agora <code>ExpectedCustomerSpend</code> . O tipo de dados mudou de um objeto para uma matriz contendo um objeto. • Atualização da conta do destinatário: o campo <code>ReceiverAccount</code> foi renomeado <code>AccountReceiver</code> . • Novo atributo do convite de engajamento: o <code>SenderContact</code> atributo foi introduzido para convites de engajamento. • <code>Customer</code> . <code>Contact</code> agora é uma	

Alteração	Descrição	Data
	matriz em vez de um objeto. OpportunityOwner , PartnerAccountManager são valores suportados como BusinessTitle . • PartnerOpportunityTeam agora é denominado OpportunityTeam . • APNPrograms é agora ApnPrograms • Os status dos convites de engajamento foram atualizados. • ListEngagementInvitations agora requer ParticipantType como RECEIVER.	

Alteração	Descrição	Data
ARCO 2	• Introduziu uma nova entidade chamada OpportunityEngagementInvitation. • Introduziu uma nova ação: ListOpportunityEngagementInvitations. • Introduziu uma nova ação: GetOpportunityEngagementInvitation. • Apresentou um novo evento: “Foi criado um convite de engajamento de oportunidade”. • AcceptOpportunity Substituído por AcceptOpportunityEngagementInvitation. • RejectOpportunity Substituído por RejectOpportunityEngagementInvitation. • Substituiu “Oportunidade aceita” por “Convite de engajamento de oportunidade aceito”. • Substituiu “Oportunidade rejeitada” por “Convite de engajamento de oportunidade rejeitado”. • PrimaryNeedsFromAWS alterado para PrimaryNeedsFromAws.	2024-08-07

Alteração	Descrição	Data
	• AWSOpportunityTeam alterado paraAwsOpportunityTeam . • AWSSalesLifeCycle alterado paraAwsSalesLifeCycle . • PrimaryNeedsFromAWSChangeReason alterado paraPrimaryNeedsFromAwsChangeReason . • AWSAccountId alterado paraAwsAccountId . • ExpectedMonthlyAWSRevenue alterado paraExpectedMonthlyAwsRevenue . • AWSFundingUsed alterado paraAwsFundingUsed . • AWSMarketplaceOffers alterado paraAwsMarketplaceOffers . • Countryalterado paraCountryCode . • PartnerOpportunityContact alterado paraPartnerOpportunityTeam . • Campo atualizado Contact.Title	

Alteração	Descrição	Data
	paraContact.BusinessTitle . • Campo atualizado o ContactInfo paraOwnerInfo . • Mudou a AWS equipe de um mapa para uma lista. • Detalhes adicionados sobre a lista de aceitos Rejection Reasons. • Forneceu informações sobre como usar o Token do Cliente. • Detalhes incluídos sobre como usar UpdateOpportunity. • Orientações adicionais sobre o uso de AWS produtos e soluções de parceiros. • Forneceu detalhes sobre a OpportunityEngagementInvitation entidade. • Atualizado StateOrRegion para incluir uma lista revisada de valores aceitos. • Implementou várias correções de bugs para melhorar o desempenho e as mensagens de erro.	

Alteração	Descrição	Data
ARCO 1	• [Interrupção] Renomeação de ApprovalStatus para ReviewStatus: Renomeamos o ApprovalStatus campo ReviewStatus para melhor refletir sua finalidade. Além disso, estamos introduzindo um novo status, Envio pendente, para indicar uma oportunidade preliminar. Agora, ReviewStatus aceitará os seguintes valores: Envio pendente, Enviado, Em análise, Aprovado, Rejeitado e Ação necessária. O status anterior de Rascunho será substituído pelo Envio pendente. • [Interrupção] Introdução da SubmitOpportunity Ação: Uma nova ação SubmitOpportunity,, foi introduzida. Ao contrário do comportamento atual em que CreateOpportunity também envia a oportunidade, agora você pode criar uma oportunidade no estado Envio pendente sem vincular imediatamente uma Solução ou AWS Produto. Para concluir o envio, use a AssociateOpportunity	2024-06-21

Alteração	Descrição	Data
	ação para vincular uma solução ou produto, seguida por SubmitOpportunity. Essa separação fornece melhor desempenho de resposta da API e uma fase de preparação flexível antes do envio. • [Quebra] Requisito de um parâmetro de catálogo e descontinuação do Sandbox Endpoint: O endpoint 'gamma' (partnercentral-selling-gamma.us-east-1.amazonaws.com) ficará indisponível após /2024. 7/30 Todas as ações da API agora exigem um parâmetro Catalog para especificar se a operação é executada no Sandbox ou no AWS catálogo. As regras de notificação agora serão filtradas pelo catálogo em vez do EnvironmentName. • [Quebra] Renomear OpportunityIdentifier para Identifier: Actions AssignOpportunity, AcceptOpportunity, e RejectOpportunity agora use Identifier em vez de OpportunityIdentifier para	

Alteração	Descrição	Data
	manter a consistência com UpdateOpportunity. • Mudanças do tipo Enum para String for APNProgram e UseCase: Estaremos convertendo CustomerUseCase, Project.APNPrograms Marketing.ActivityUseCases, e Project.CustomerUseCase do tipo de enumeração para o tipo de string para reduzir os riscos associados à alteração de valores. Esses campos aceitarão valores somente de listas predefinidas, mas não estão disponíveis de forma nativa nos SDKs. • Tratamento aprimorado de erros: o tratamento de erros nas APIs foi significativamente aprimorado para facilitar a depuração e a resolução mais rápida de problemas. O novo tratamento de erros estrutura os erros em dois estágios: 1. Falha na validação da solicitação: verifica os tipos e o formato dos dados, ou 2. Falha na validação comercial: todos os problemas em	

Alteração	Descrição	Data
	uma carga serão listados em uma resposta, especificando o campo que contém erros. Esse feedback consolidado permite que você solucione e corrija erros com mais eficiência. Códigos e formatos de erros foram padronizados. - Identificador de origem do cliente: capacidade dos parceiros incluírem o X-Amzn-User-Agent cabeçalho em suas solicitações, o que ajuda a identificar a origem do tráfego. Use o formato [Nome da solução, incluindo provedor da solução] [Versão] Por exemplo: AWS Partner CRM Connector v2.0.1 - Correção de bug, já implantada, sem alteração no SDK] Criação da oportunidade e ID AWS da conta: os parceiros agora podem iniciar uma oportunidade enquanto alteram a ID da AWS conta de nula para um valor válido sem encontrar erros. Anteriormente, um bug impedia que os parceiros realizassem	

Alteração	Descrição	Data
	essas ações separadamente. • Correção de bug, já implantada, sem alteração no SDK] Associação de soluções com novas oportunidades: os parceiros podem associar uma solução a uma oportunidade recém-criada. Antes, quando os parceiros criavam uma nova oportunidade e associavam simultaneamente uma solução, as informações da solução estavam sendo perdidas. Esse problema foi resolvido. • Correção de bug, já implantada, sem alteração no SDK] Exibição dos detalhes do proprietário da oportunidade: o problema em que algumas oportunidades exibiam os detalhes do proprietário em um campo combinado LastName "" em vez dos campos esperados de "E-mail", "FirstName" e "LastName" foi corrigido. • Correção de bug, já implantada, sem alteração no SDK] Customer.Account.WebsiteUrl Campo opcional: o Customer.	

Alteração	Descrição	Data
	Account.WebsiteUrl campo se tornou opcional quando o AccessControls.NationalSecurity campo é “Sim”, alinhando assim o comportamento da API com o comportamento da interface de usuário existente. Se o AccessControls.NationalSecurity campo for “Não” ou nulo, o WebsiteUrl campo será obrigatório como uma validação comercial. • Correção de bug, já implantada, sem alteração no SDK] Requisito de ID de AWS conta para parceiros de consultoria: a inconsistência na interface do usuário que mostrava o campo ID da AWS conta como opcional para parceiros de consultoria foi resolvida. Agora é um campo obrigatório condicional, conforme a documentação. • Correção de bug. Para ser implantado 6/19, nenhuma alteração no SDK] Estágio de uma Co-sell oportunidade para fechada e perdida: os parceiros	

Alteração	Descrição	Data
	agora podem atualizar com sucesso o estágio de uma Co-sell oportunidade para encerrada e perdida, desde que um motivo fechado seja fornecido. Isso resolve o erro anterior dizendo “Campo obrigatório ausente: o motivo da perda é obrigatório ao fechar a oportunidade”. • Documentação] Melhorias na documentação: Fiz várias melhorias na documentação para refletir as mudanças. • Problemas conhecidos com uma correção pendente e uma lista dos próximos recursos: • Ao executar Associate Opportunity ações com soluções ou produtos, as mensagens de erro das Ofertas são mostradas incorretamente. • Capacidade de criar oportunidades remotamente e simular o processo de AWS validação no Sandbox • O código postal retorna nulo quando está no	

Alteração	Descrição	Data
	formato XXXXX-XXXX dos EUA. • As próximas etapas e o histórico da próxima etapa não estão disponíveis para oportunidades marcadas como “Não preciso de suporte de”AWS. • Não é possível obter oportunidades históricas com descrições de problemas comerciais do cliente com mais de 255 caracteres. • Capacidade de excluir oportunidades que estão pendentes de envio. • LeadSource não está disponível no modelo de dados.	

Alteração	Descrição	Data
Versão beta 5	• Alterado LifeCycle.Approval Status para LifeCycle .ReviewStatus • Alterado Insights. ReviewComments para LifeCycle.ReviewComments • Adicionado LifeCycle .ReviewStatusReason (novo atributo) • Novo valor adicionado “Pendente” a PartnerAcceptance.Status • SDKs lançados para dotnet, java, javascript, python, ruby. • Guia de referência da API atualizado para refletir as alterações acima.	2024-04-19

Alteração	Descrição	Data
Versão beta 4	• Teste de sandbox adicionado: capacidade de testar a notificação de evento de oportunidade no sandbox. • Documentação adicionada: introduziu exemplos de regras de notificação de eventos de oportunidade. • Lançamentos de SDK adicionados: SDKs lançados para Java (V2), Javascript (V2), (V3) e Python DotNet (V3). • Formato de data atualizado: atualizado para seguir os padrões ISO.	2024-03-04

Alteração	Descrição	Data
Versão beta 3	• Arquivo de registro de alterações adicionado: introduziu um arquivo de registro de alterações para melhor rastreamento de atualizações e alterações. • Guia de referência de API atualizado (PDF): lançou uma nova versão do Guia de referência de API em formato PDF. • Guia de referência de API atualizado (versão da Web): a versão web do Guia de referência de API foi atualizada. • DotNet SDK adicionado: Lançamos o DotNet SDK, expandindo nosso suporte para diferentes ambientes de programação. • SDK do Python Boto 3 atualizado (V2): Lançada a versão 2 do SDK do Python Boto 3, introduzindo novos recursos e melhorias.	2024-01-24

Alteração	Descrição	Data
Versão beta 2	• Guia de referência de API adicionado (versão Web): Lançou uma nova versão do Guia de referência de API (versão Web). Nota: Esta versão requer que um servidor web local seja executado após a extração para uma funcionalidade adequada.	2024-01-07
Versão beta 1	• Boto3 adicionado (versão inicial): lançou a versão inicial do Boto3, um SDK Python para nossos serviços. • Guia do programa beta adicionado: introduzimos um guia para os participantes do nosso programa de testes beta. • Guia de referência de API adicionado (PDF): Carregamos a primeira versão do nosso Guia de referência de API em formato PDF, fornecendo documentação abrangente para nossa API.	2023-12-15

Alteração	Descrição	Data
Versão beta 0	Essa versão inicial fornece um conjunto de funcionalidades para gerenciar e otimizar engajamentos e oportunidades de parceiros no AWS Partner Central.	2023-11-15

Histórico do documento

Veja a seguir uma lista dos lançamentos da documentação para essa referência de API.

Alteração	Descrição	Data
Documentação adicionada para APIs de medição de receita de parceiros	AWS O Partner Central adicionou novas APIs para a Medição da Receita de Parceiros (PRM).	30 de junho de 2026
Documentação atualizada para ACE Resonate	AWS Resposta atualizada da API da Partner Central para GetAwsOpportunitySummary ações de suporte ao Cosell Motion.	16 de junho de 2026
Documentação adicionada para o Partner Central MCP Server	AWS O servidor MCP dos agentes centrais de parceiros fornece ferramentas centrais de parceiros por meio do Model Context Protocol (MCP).	16 de março de 2026
Lançamento do RC 5	Lançada a versão Candidate 5 da API do AWS Partner Central. Introduziu uma nova API de conta no AWS Partner Central, permitindo que os parceiros gerenciem as informações da conta, o registro, os perfis, o gerenciamento de domínio e as conexões de parceiros. Introduziu uma nova API de benefícios no AWS Partner Central, permitindo que	30 de novembro de 2025

os parceiros descubram, se inscrevam e gerenciem benefícios em todos os programas de AWS parceiros por meio de uma interface centralizada. Ações aprimoradas de engajamento da API de vendas para apoiar o fluxo de trabalho de gerenciamento de leads. Foram adicionados guias de uso abrangentes e documentação de API para APIs de contas e benefícios com fluxos de trabalho detalhados e padrões de integração. Documentação aprimorada da API de vendas para gerenciamento de engajamento com contexto de leads.

Lançamento do RC 4

Lançada a versão Candidate 4 da API do AWS Partner Central. Introduziu a API de canais para gerenciar relacionamentos entre canais e colaborações entre parceiros, permitindo que os parceiros se qualifiquem para os benefícios do programa ao realizar transações com clientes finais usando o Billing Transfer. A documentação foi reorganizada para oferecer suporte a várias APIs do AWS Partner Central.

19 de novembro de 2025

Documentação adicionada para oportunidades com vários parceiros	Multi-partner opportunities (em versão prévia) é uma experiência integrada no AWS Partner Central que permite que AWS os parceiros encontrem e se conectem com outros parceiros e vendam juntos ofertas com clientes.	4 de dezembro de 2024
Lançamento do RC 3	Lançada a versão Candidate 3 da API AWS Partner Central	17 de outubro de 2024
Lançamento do RC 2	Lançada a versão Candidate 2 da API AWS Partner Central	7 de agosto de 2024
Versão RC 1	Lançada a versão Candidate 1 da API AWS Partner Central	21 de junho de 2024
Lançamento do Beta 5	Lançada a versão Beta 5 da API AWS Partner Central	19 de abril de 2024
Versão beta 4	Lançada a versão beta 4 da API AWS Partner Central	4 de março de 2024
Lançamento do Beta 3	Lançada a versão Beta 3 da API AWS Partner Central	24 de janeiro de 2024
Versão beta 2	Lançada a versão Beta 2 da API AWS Partner Central	7 de janeiro de 2024
Versão beta 1	Lançada a versão beta 1 da API AWS Partner Central	15 de dezembro de 2023
Lançamento do Beta 0	AWS Versão inicial da API Partner Central	15 de novembro de 2023