AWS EBS Snapshots with IAM Cross-Account Access

3/5/2014

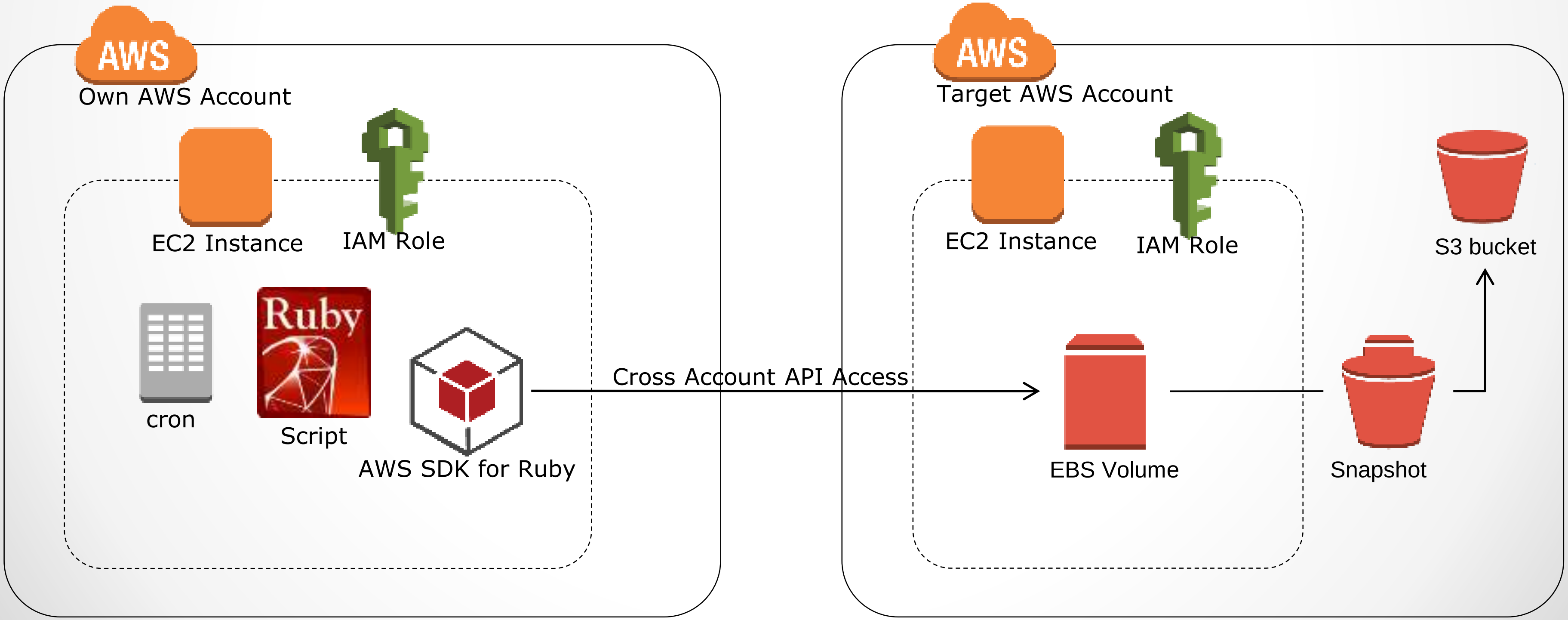
Naoya Hashimoto

Table of Contents

- Overview
- IAM Cross-Account Access
- IAM Cross-Account Access Setup Process
- IAM Policy for EBS Snapshot
- IAM Policy for AssumeRole
- How to set up IAM Cross-Account Access (1-8)
- EBS Snapshots environment Setup Process
- EBS Snapshots script Overview
- How to set up EBS snapshot environment (1-2)
- CloudFormation template for EBS snapshots Environment

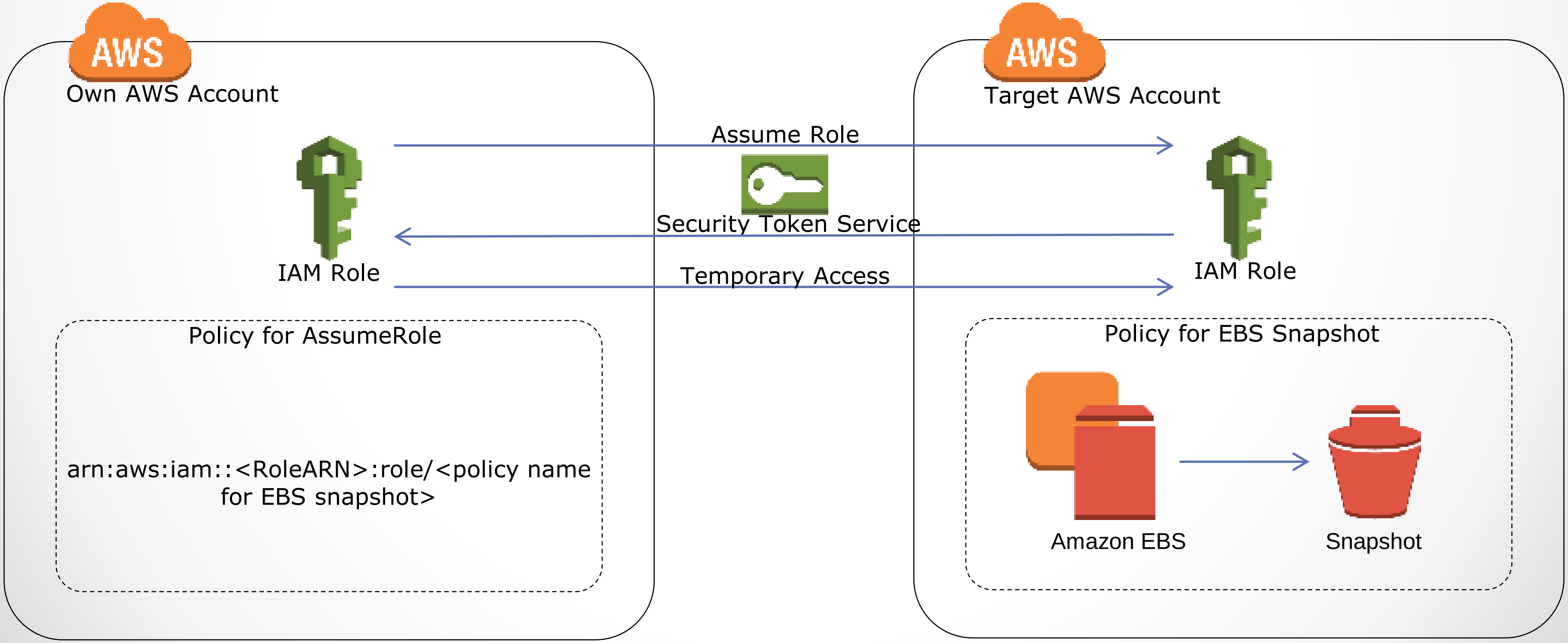
Overview

- Take EBS Snapshots for EBS volumes to retrieve other AWS account's EBS snapshot over IAM Cross Account Access.
- Integrate EC2 instance with Amazon Linux AMI and install Ruby script with AWS SDK for Ruby to take EBS Snapshots.
- Use IAM Role to assume the role to take EBS snapshot for Cross Account API Access.



IAM Cross-Account Access

- Use the AWS STS(Security Token Service) to request limited-privilege credentials for AWS IAM user
- Create IAM Role to allow to manage EBS snapshot.
- Use Assume Role for cross-account access.

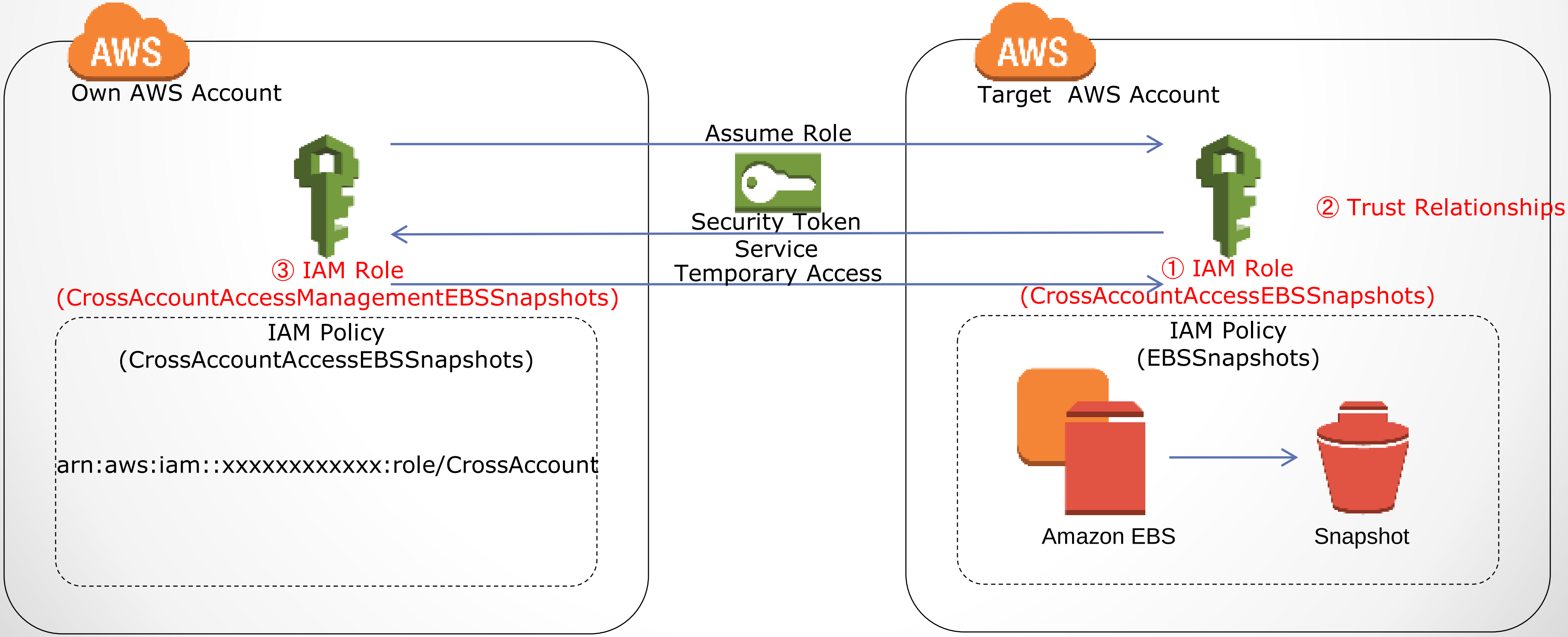


IAM Cross-Account Access Setup Process

① Create IAM Role and apply IAM Policy for EBS Snapshot on the target AWS account
- Role Name: CrossAccountAccessEBSSnapshots
- Policy Name: EBSSnapshots
- Policy Document: See P7

② Establish the trust relationships between the IAM Role and own AWS account.

③ Create IAM Role to request STS (Security Token Service) for Cross-Account Access on own AWS account
- Role Name: CrossAccountAccessManagementEBSSnapshots
- Policy Name: CrossAccountAccessEBSSnapshots
- Policy Document: See P8



IAM Policy for EBS Snapshot

Policy Name: EBSSnapshots

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1391473701000",
      "Effect": "Allow",
      "Action": [
        "ec2:Describe*",
        "ec2:*Tags",
        "ec2:CopySnapshot",
        "ec2:CreateSnapshot",
        "ec2:DeleteSnapshot",
        "ec2:ModifySnapshotAttribute",
        "ec2:ResetSnapshotAttribute"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

IAM Policy for AssumeRole

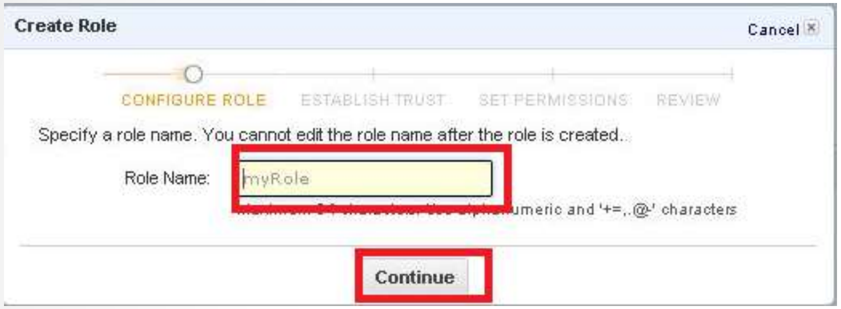
Policy Name: CrossAccountAccessEBSSnapshots

```
{
 "Statement" : [
  {
    "Effect" : "Allow",
    "Action" : "sts:AssumeRole",
    "Resource" : "arn:aws:iam::<Target AWS Account ID *1>:role/EBSSnapshots"
  }
 ]
}
```

*1 Put the ID of target AWS Account on <Target AWS Account ID>

How to set up IAM Cross-Account Access (1)

- Log in to AWS Management Console of the target AWS Account.
- Click [Services] - [Deploy & Management] – [IAM] and get to the IAM dashboard.
- Click [Roles] – [Create New Role].
- Input [Role Name] and click [Continue].
  - Role Name: CrossAccountEBSSnapshot

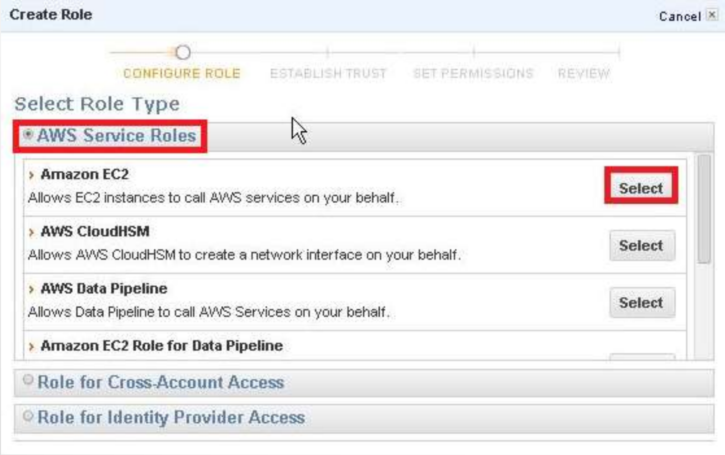


Specify a role name. You cannot edit the role name after the role is created.

Role Name: myRole

Maximum 64 characters. Use alphanumeric and '+=,.@-' characters

- Select [AWS Service Roles] and [Amazon EC2 Select].



› Amazon EC2
Allows EC2 instances to call AWS services on your behalf.

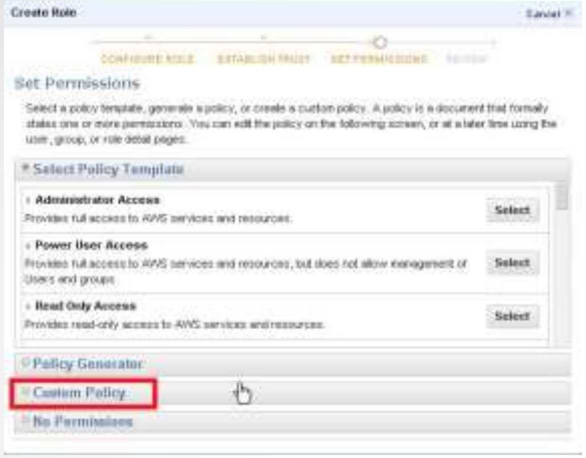
› AWS CloudHSM
Allows AWS CloudHSM to create a network interface on your behalf.

› AWS Data Pipeline
Allows Data Pipeline to call AWS Services on your behalf.

› Amazon EC2 Role for Data Pipeline

How to set up IAM Cross-Account Access (2)

- Select [Custom Policy].



Create Role

Set Permissions

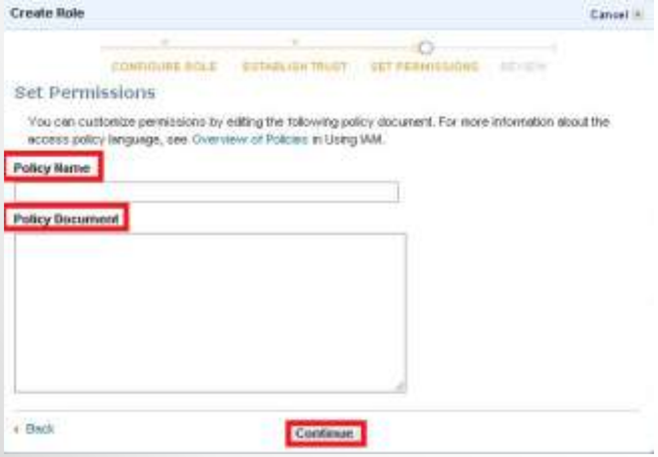
Select a policy template, generate a policy, or create a custom policy. A policy is a document that formally states one or more permissions. You can edit the policy on the following screen, or at a later time using the user, group, or role detail pages.

Administrator Access
Provides full access to AWS services and resources.

Power User Access
Provides full access to AWS services and resources, but does not allow management of Users and groups.

Read Only Access
Provides read-only access to AWS services and resources.

- Input the following items and click [Continue].
  - Policy Name: EBSSnapshots
  - Policy Document: See P7



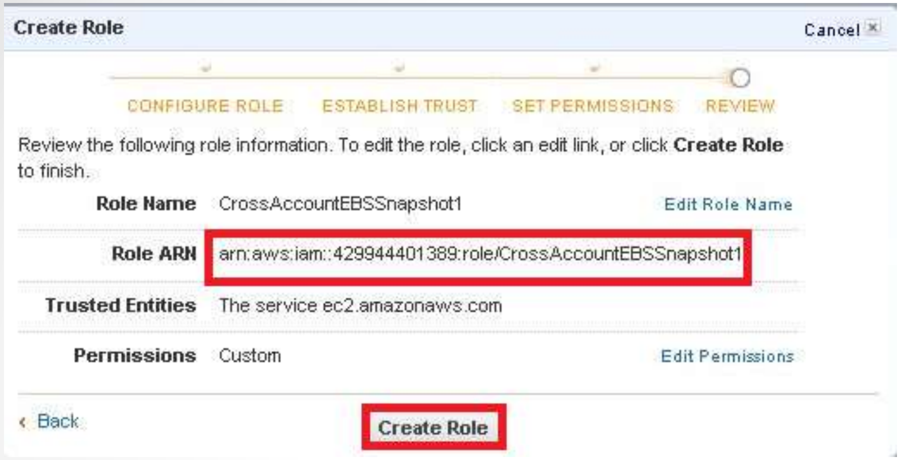
Create Role

Set Permissions

You can customize permissions by editing the following policy document. For more information about the access policy language, see Overview of Policies in Using IAM.

How to set up IAM Cross-Account Access (3)

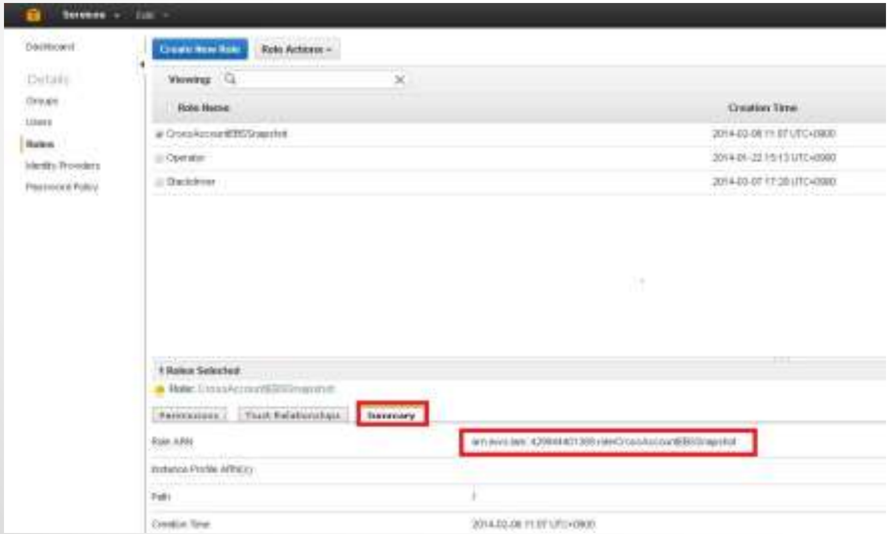
- Remember or Copy [Role ARN] and Click [Create Role].



Review the following role information. To edit the role, click an edit link, or click **Create Role** to finish.

Role Name	CrossAccountEBSSnapshot1
Role ARN	arn:aws:iam::429944401389:role/CrossAccountEBSSnapshot1
Trusted Entities	The service ec2.amazonaws.com
Permissions	Custom

* You can check [Role ARN] on the summary section after creating IAM Role



Role Name	Creation Time
CrossAccountEBSSnapshot	2014-02-08 11:07 UTC+0900
Operator	2014-01-22 15:13 UTC+0900
Stackdriver	2014-03-07 17:20 UTC+0900

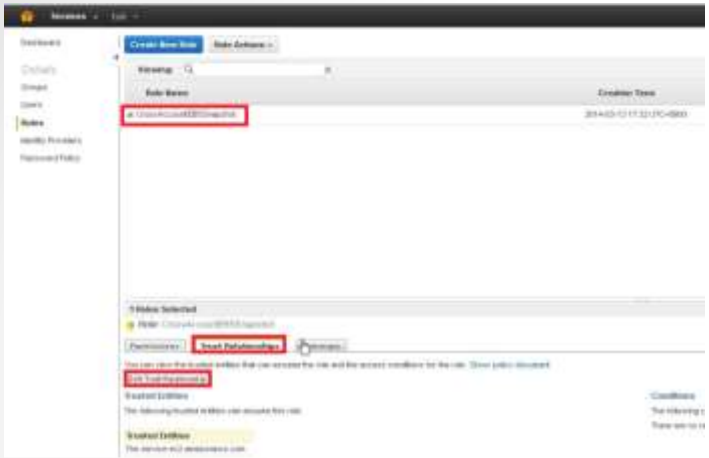
1 Roles Selected

Role: CrossAccountEBSSnapshot

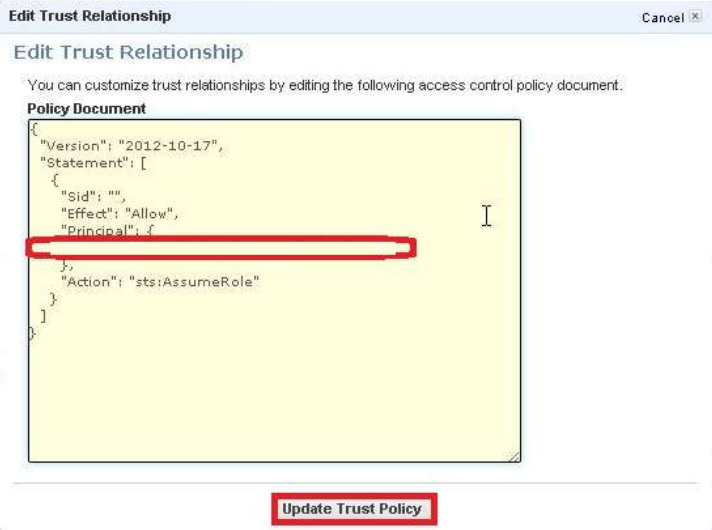
Role ARN	arn:aws:iam::429944401389:role/CrossAccountEBSSnapshot
Instance Profile ARN(s)	
Path	/
Creation Time	2014-02-08 11:07 UTC+0900

How to set up IAM Cross-Account Access (4)

- Click [Role ARN] – [Trust Relationships] – [Edit Trust Relationship].



- Modify the “Principal” section of the policy as follows and click [Update Trust Policy].



Edit Trust Relationship

You can customize trust relationships by editing the following access control policy document.

Policy Document

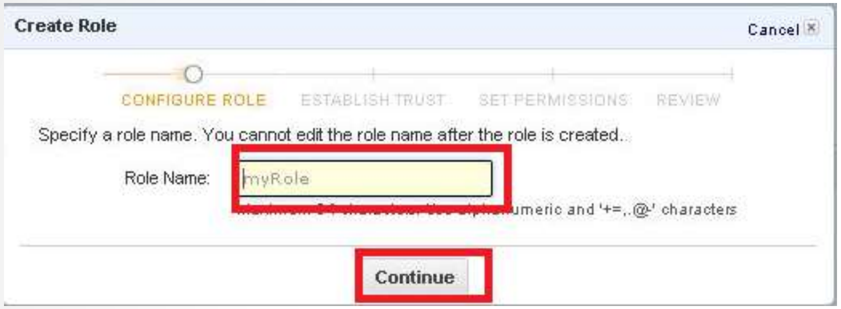
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {

      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

```
"Principal": {
  "Service": “ec2.amazonaws.com"},
↓
},
"Principal": {
  "AWS": "arn:aws:iam::<Own AWS Account ID>:root"
},
```

How to set up IAM Cross-Account Access (5)

- Log in to AWS Management Console of own AWS Account.
- Click [Services] - [Deploy & Management] – [IAM] and get to the IAM dashboard.
- Click [Roles] – [Create New Role].
- Input [Role Name] and click [Continue].
  - Role Name: CrossAccountAccessManagementEBSSnapshots

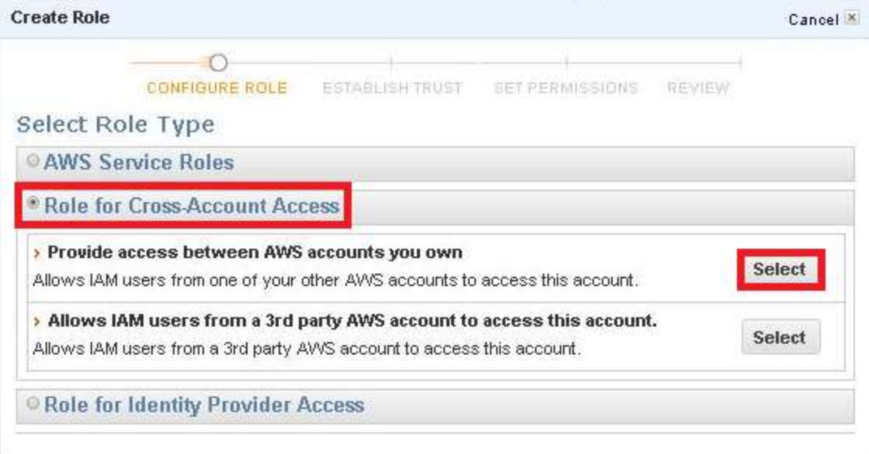


Specify a role name. You cannot edit the role name after the role is created.

Role Name: myRole

Maximum 64 characters. Use alphanumeric and '+=,.@-' characters

- Select [Role for Cross-Account Access] and click [Provide access… Select].



Select Role Type

○ AWS Service Roles

● Role for Cross-Account Access

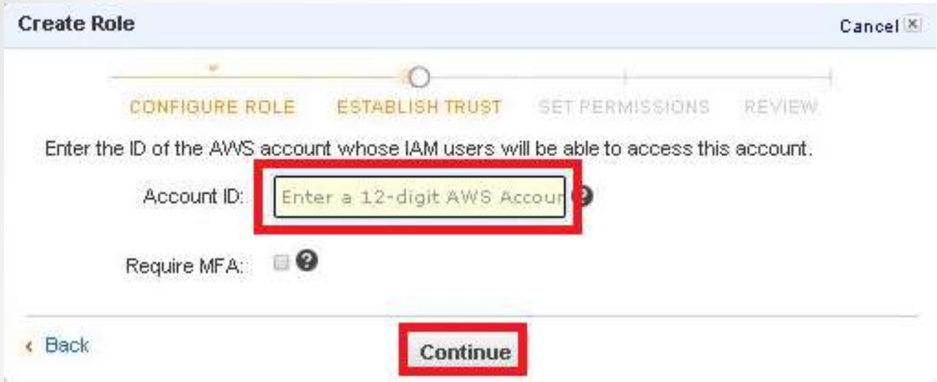
› Provide access between AWS accounts you own
Allows IAM users from one of your other AWS accounts to access this account.

› Allows IAM users from a 3rd party AWS account to access this account.
Allows IAM users from a 3rd party AWS account to access this account.

○ Role for Identity Provider Access

How to set up IAM Cross-Account Access (6)

- Input the following item and click [Continue].
  Account ID: <Target AWS Account ID>



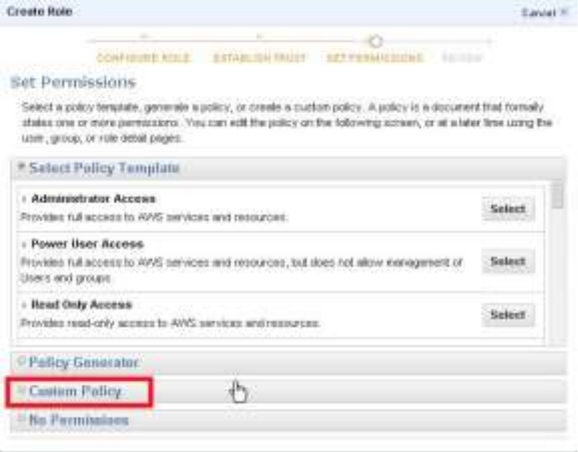
Enter the ID of the AWS account whose IAM users will be able to access this account.

Account ID: Enter a 12-digit AWS Accour

Require MFA: □

How to set up IAM Cross-Account Access (7)

- Select [Custom Policy].



Create Role

Set Permissions

Select a policy template, generate a policy, or create a custom policy. A policy is a document that formally states one or more permissions. You can edit the policy on the following screen, or at a later time using the user, group, or role detail pages.

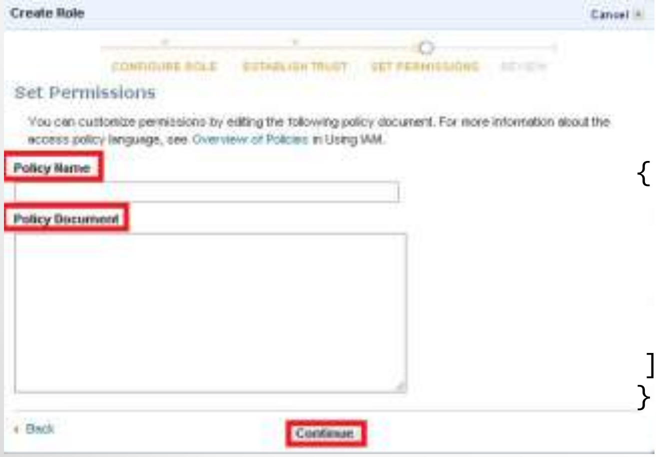
Select Policy Template

Administrator Access
Provides full access to AWS services and resources.

Power User Access
Provides full access to AWS services and resources, but does not allow management of Users and groups.

Read Only Access
Provides read-only access to AWS services and resources.

- Input the following items and click [Continue]. * Replace<AWS Account ID> with target AWS account ID
  - Policy Name: CrossAccountAccessEBSSnapshots
  - Policy Document: See P8



Create Role

Set Permissions

You can customize permissions by editing the following policy document. For more information about the access policy language, see Overview of Policies in Using IAM.

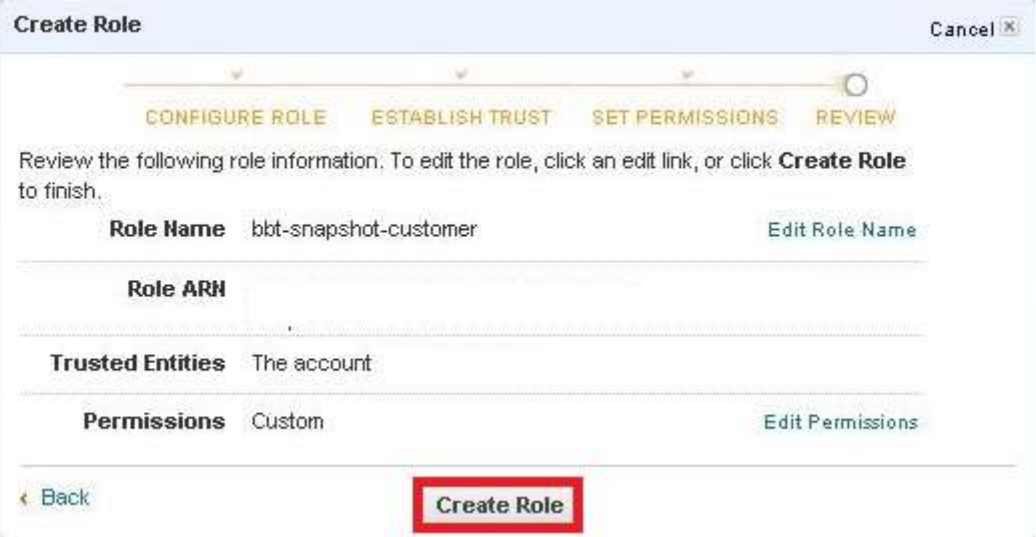
Policy Name

Policy Document

```
{ "Statement" : [
  {
    "Effect" : "Allow",
    "Action" : "sts:AssumeRole",
    "Resource" : "arn:aws:iam::<AWS Account ID>:role/CrossAccountEBSSnapshot"
  }
 ]
}
```

How to set up IAM Cross-Account Access (8)

- Click [Create Role].



Create Role

Review the following role information. To edit the role, click an edit link, or click **Create Role** to finish.

Role Name bbt-snapshot-customer

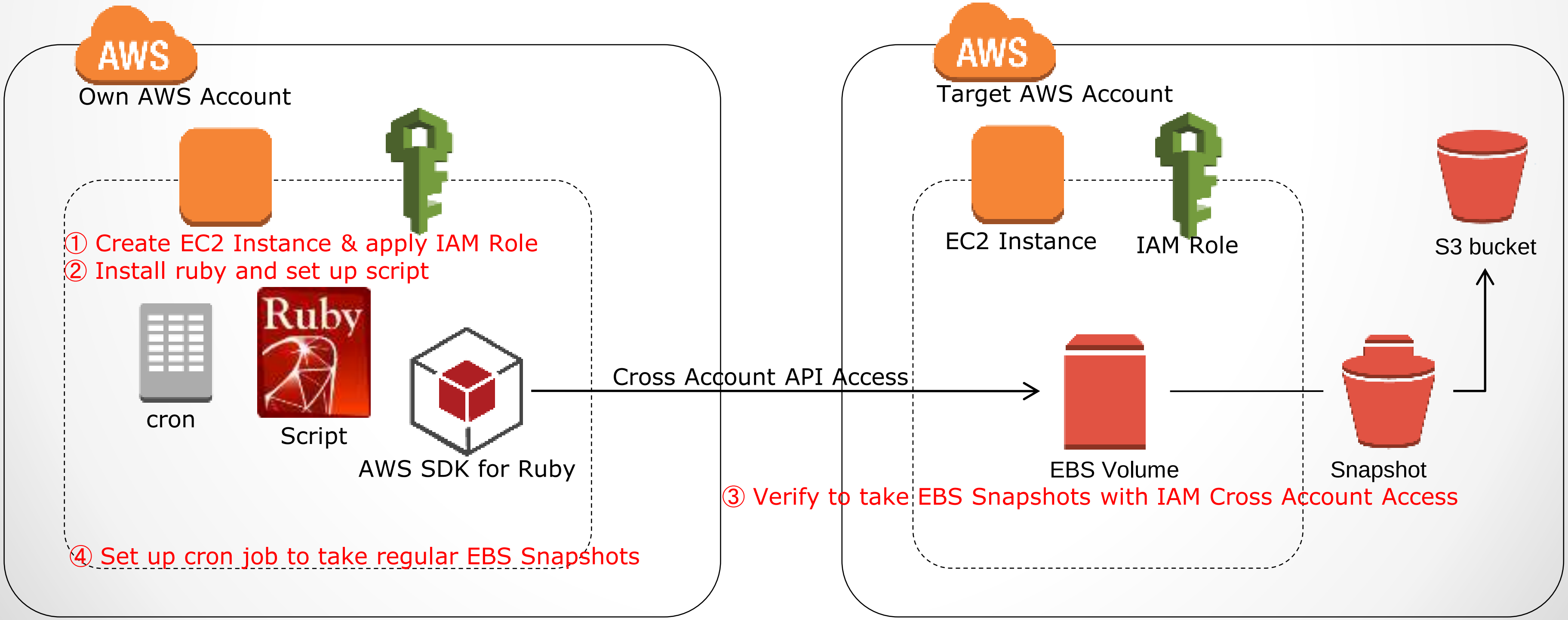
Role ARN

Trusted Entities The account

Permissions Custom

EBS Snapshots environment Setup Process

① Create EC2 Instance and attach IAM Role for Cross-Account Access
② Install ruby, gems (aws-sdk) and set up ruby script
③ Verify to take EBS snapshots for
④ Set up cron jobs to create EBS snapshot


AWS
Own AWS Account
① Create EC2 Instance & apply IAM Role
② Install ruby and set up script
Ruby
cron
Script
AWS SDK for Ruby
④ Set up cron job to take regular EBS Snapshots
Cross Account API Access
AWS
Target AWS Account
EC2 Instance
IAM Role
S3 bucket
EBS Volume
Snapshot
③ Verify to take EBS Snapshots with IAM Cross Account Access

EBS Snapshots script Overview

Cron(ec2-user)

```
$HOME/bin/create_ebs_snapshot.sh <EBS Volume ID> <generation *1> <Role ARN>
```

*1 Every EBS snapshots maintains by default without the number of generation

create_ebs_snapshot.sh

```
volume_id=$1
generation=$2
role_arn=$3
$HOME/.rvm/rubies/${ruby_ver}/bin/ruby ./create_ebs_snapshot_crossaccount.rb -v $volume_id -g $generation -r $role_arn \
>> ${logfile} 2>&1
```

create_ebs_snapshot_crossaccount.rb

```
# Create EBS snapshot
ec2 = AWS::EC2.new
reg = ec2.regions[endpoint]
snapshot = reg.volumes[volume_id].create_snapshot(description)
sleep 1 until [:completed, :error].include?(snapshot.status)
snapshot.add_tag('Name', :value => name)

# Describe snapshot status
puts "#{name} Snapshot iD: #{snapshot.id}, Progress: #{snapshot.progress}%, Status: #{snapshot.status}"

# Rotate and Delete EBS snapshot
if generation
  snapshots = reg.snapshots.filter('volume-id', volume_id).sort_by { |x| x.start_time }.reverse
  ss = snapshots[generation..-1]
  ss.each { |x| x.delete } unless ss.nil?
end
```

Packages to run script

- autoconf
- automake
- aws-sdk
- bison
- gcc
- gcc-c++
- git
- jq
- libffi-devel
- libtool
- libxml2-devel
- libxslt-devel
- libyaml-devel
- make
- openssl-devel
- patch
- readline-devel
- ruby-2.0.0
- rvm

How to set up EBS snapshot environment (1)

- Install libraries

```
$ sudo yum -y groupinstall "Development libraries" "Development tools"
$ sudo yum -y install git libxml2-devel libxslt-devel
```

- Install RVM, Ruby, gems, aws sdk for ruby

```
$ \curl -L https://get.rvm.io | bash -s stable
$  . ~/.bashrc
$ rvm install 2.0.0
$ rvm use 2.0.0 –default
$ gem i aws-sdk
```

- Install jq

```
$ git clone https://github.com/stedolan/jq.git
$ cd jq
$ autoreconf -i
$ ./configure && make && sudo make install
```

- Set up script

```
$ git clone https://github.com/hashnao/aws/archive/master.zip $HOME/bin
$ chown -R ec2-user:ec2-user $HOME/bin
$ find $HOME/bin -type f -name "*.sh" -exec chmod 755 {} \;
```

- Set up crontab * sample job

```
$ crontab -e
10 2 * * * $HOME/bin/ebssnapshot/1.0/create_ebs_snapshot_crossaccount.sh <Volume ID>
<generation><RoleARN>
```

How to set up EBS snapshot environment (2)

- Verify to take EBS snapshots with IAM Cross-Account Access

```
$ $HOME/bin/ebssnapshot/1.0/create_ebs_snapshot_crossaccount.sh <EBS Volume ID> <generation> <Role ARN>
```

Ex.

```
$ $HOME/bin/create_ebs_snapshot_crossaccount.sh vol-xxxxxxxx 1 arn:aws:iam::<AWS Account ID>:role/CrossAccountEBSSnapshot
```

- See the log file to confirm the EBS snaphost has been taken.

```
$ tail -f $HOME/log/create_ebs_snapshot_crossaccount.sh_<yyyymmdd>.log
vol-3967a333-2014/03/13_18:03:45 Snapshot iD: snap-325ab6d3, Progress: 100%, Status: completed
```

* The following messages are output if the IAM Role of EBS Snapshot is not correct.

```
/home/ec2-user/.rvm/gems/ruby-2.0.0-p353/gems/aws-sdk-1.33.0/lib/aws/core/client.rb:374:in `return_or_raise': User: arn:aws:sts::<AWS Account ID>:assumed-role/bbt-snapshot/i-47409c40 is not authorized to perform: sts:AssumeRole on resource: arn:aws:iam::099897076573:role/CrossAccountEBSSnapshot (AWS::STS::Errors::AccessDenied)
        from /home/ec2-user/.rvm/gems/ruby-2.0.0-p353/gems/aws-sdk-1.33.0/lib/aws/core/client.rb:475:in `client_request'
        from (eval):3:in `assume_role'
        from /home/ec2-user/.rvm/gems/ruby-2.0.0-p353/gems/aws-sdk-1.33.0/lib/aws/sts.rb:58:in `assume_role'
        from ./create_ebs_snapshot_crossaccount.rb:48:in `<main>'
```

CloudFormation template for EBS snapshots Environment

AWS CloudFormation

- Template PATH on github

https://github.com/hashnao/aws-cloudformation/blob/master/EC2/ec2-ebssnapshot.template

- Validate template

```
$ wget https://raw.github.com/hashnao/aws-cloudformation/master/EC2/ec2-ebssnapshot-instance.template
$ aws cloudformation validate-template \
--template-body file://$PWD/ec2-ebssnapshot-instance.template
```

- Deploy stack

```
$ aws cloudformation create-stack \
--capabilities CAPABILITY_IAM \
--template-body file://$PWD/ec2-ebssnapshot-instance.template \
--parameters \
ParameterKey=InsntanceType,ParameterValue=<InstanceType> \
ParameterKey=KeyName,ParameterValue=<SSHKeyName> \
ParameterKey=SSHLocatoin,ParameterValue=<CIDR> \
--stack-name <StackName>
```

Never fail to regularly take EBS snapshot just in case.

…