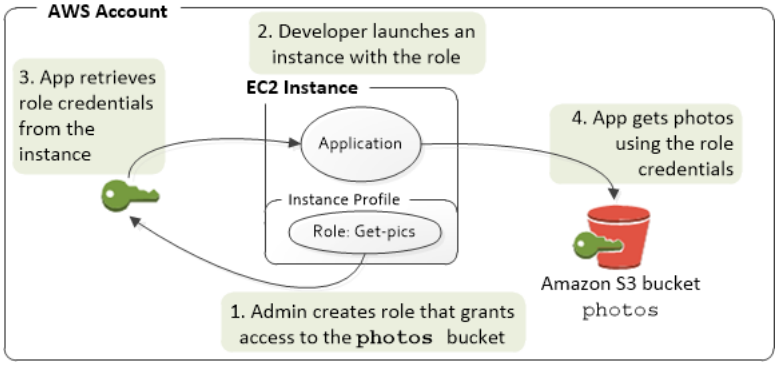
IAM role credentials

Applications that run on an EC2 instance must include AWS credentials in their AWS API requests. You could store AWS credentials directly within the EC2 instance and allow applications in that instance to use those credentials. But you would then have to manage the credentials and ensure that they securely pass the credentials to each instance and update each EC2 instance when it's time to rotate the credentials. That's a lot of additional work. Instead, you can and should use an IAM role to manage temporary credentials for applications that run on an EC2 instance. When you use a role, you don't have to distribute long-term credentials (such as a user name and password or access keys) to an EC2 instance.



Instead, the role supplies temporary permissions that applications can use when they make calls to other AWS resources. When you launch an EC2 instance, you specify an IAM role to associate with the instance. Applications that run on the instance can then use the role-supplied temporary credentials to sign API requests.

When the application runs, it obtains temporary security credentials from Amazon EC2 instance metadata. These are temporary security credentials that represent the role and are valid for a limited period of time to access various AWS resources. You can fetch the temporary security credentials from the instance by requesting it from this endpoint:

curl http://169.254.169.254/latest/meta-data/iam/security-credentials/s3access

In this particular scenario, you have to use a combination of IAM Role and EC2 instance metadata to provide the web application the required access it needs to access the S3 bucket. Hence, the correct answer is the following option:

1. Create an IAM role with a policy that allows listing and uploading of the objects in the S3 bucket. Launch the EC2 instance with the IAM role.

2. Program your web application to retrieve the temporary security credentials from the EC2 instance metadata.

References:

http://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_use_switch-role-ec2.html

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html#instance-metadata-security-credentials

Check out this AWS IAM Cheat Sheet:

https://tutorialsdojo.com/aws-cheat-sheet-aws-identity-and-access-management-iam/

AWS - STS (Temporary Credentials)

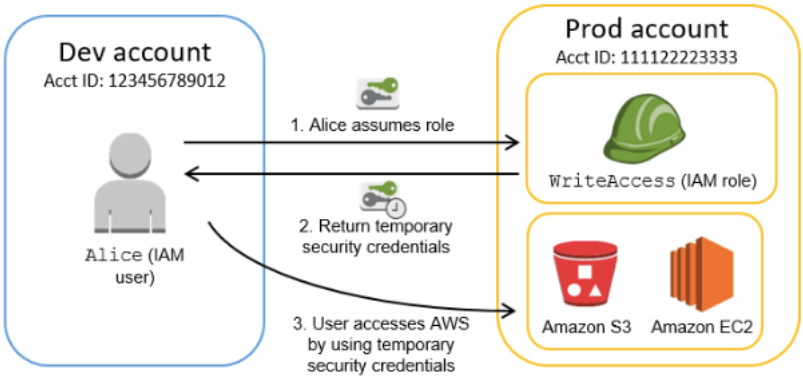
Temporary Credentials
https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_temp.html

AWS Security Token Service (AWS STS) is the service that you can use to create and provide trusted users with temporary security credentials that can control access to your AWS resources.

Temporary security credentials work almost identically to the long-term access key credentials that your IAM users can use.

In this diagram, IAM user Alice in the Dev account (the role-assuming account) needs to access the Prod account (the role-owning account). Here's how it works:

- Alice in the Dev account assumes an IAM role (WriteAccess) in the Prod account by calling AssumeRole.
- STS returns a set of temporary security credentials.
- Alice uses the temporary security credentials to access services and resources in the Prod account.
- Alice could, for example, make calls to Amazon S3 and Amazon EC2, which are granted by the WriteAccess role.

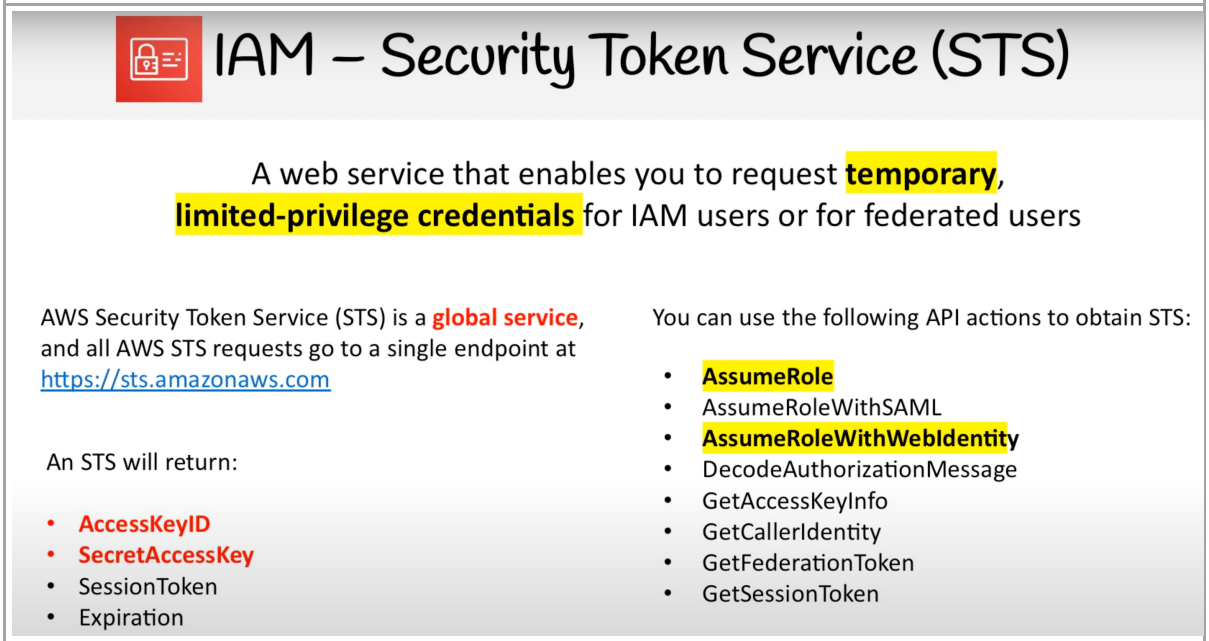


You are designing a photo-sharing mobile app for an advertising company. The app will store all pictures directly uploaded by users in a single Amazon S3 bucket and users will also be able to view and download their own pictures directly from the Amazon S3 bucket. You are to configure security on the

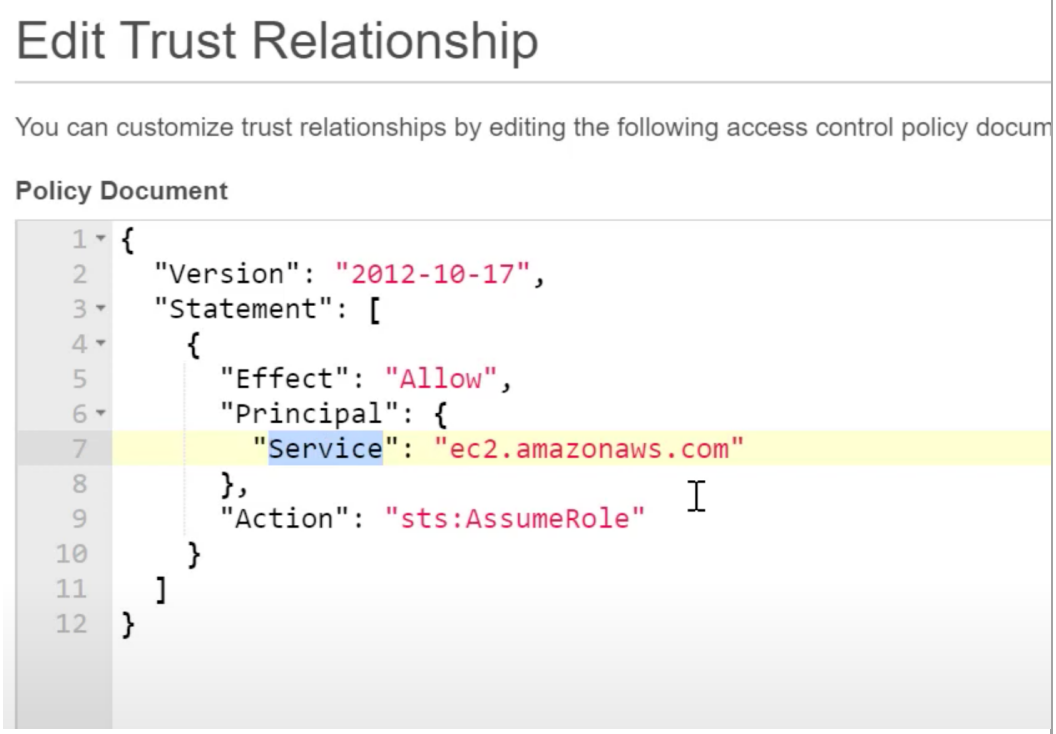
application to handle potentially millions of users in the most secure manner possible.

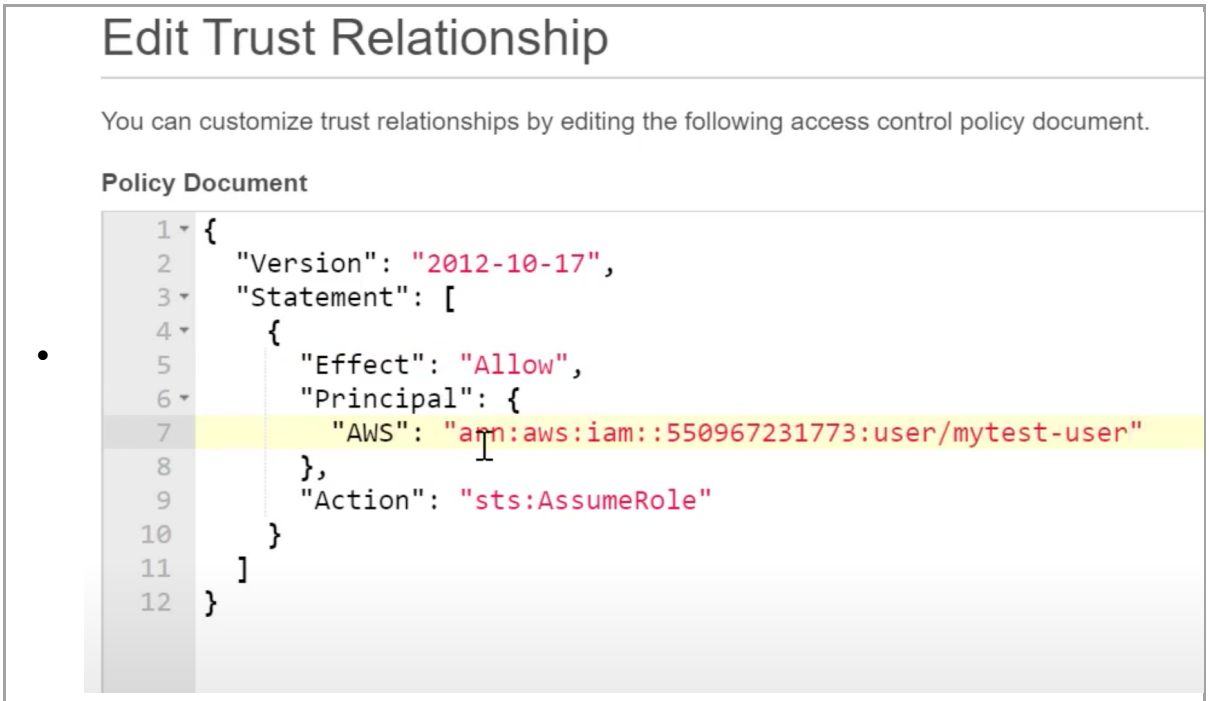
How do you set up the user registration flow in AWS for this mobile app?

- Store user information in Amazon RDS and create an IAM Role with appropriate permissions. Generate new temporary credentials using the AWS Security Token Service 'AssumeRole' function every time the user uses their mobile app and creates new temporary credentials. These credentials will be stored in the mobile app's memory and will be used to access Amazon S3.



- Create IAM user
- Create IAM rule and Edit Trust Relationship



- 

- Get the credentials, using STSRole API
  - Linux Script - example
    - Using temporary security credentials with the AWS SDKs

      assumeRoleResult = AssumeRole(role-arn);
      tempCredentials = new SessionAWSCredentials(
        assumeRoleResult.AccessKeyId,
        assumeRoleResult.SecretAccessKey,
        assumeRoleResult.SessionToken);
      s3Request = CreateAmazonS3Client(tempCredentials);
    - Using temporary security credentials with the AWS CLI
    - `aws sts assume-role --role-arn arn:aws:iam::123456789012:role/`*`role-name`* `--role-session-name "RoleSession1" --profile` *`IAM-user-name`* `> assume-role-output.txt`
    - https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_temp_use-resources.html

      https://ripon-banik.medium.com/aws-assume-role-script-4e7d3b548f20

      SCRIPT:

```
role_arn='arn:aws:iam::account_id:role/role-test'
role_session_name='test'
profile_name='test'
temp_role=$(aws sts assume-role \
    --role-arn $role_arn \
    --role-session-name $role_session_name)
export AWS_ACCESS_KEY_ID=$(echo $temp_role | jq -r
.Credentials.AccessKeyId)
export AWS_SECRET_ACCESS_KEY=$(echo $temp_role | jq -r
.Credentials.SecretAccessKey)
```

```
export AWS_SESSION_TOKEN=$(echo $temp_role | jq -r
.Credentials.SessionToken)
aws configure set aws_access_key_id $AWS_ACCESS_KEY_ID --profle
$profile_name
aws configure set aws_secret_access_key $AWS_SECRET_ACCESS_KEY --
profile $profile_name
aws configure set aws_session_token $AWS_SESSION_TOKEN --profle
$profile_name
```

SCRIPT - with MFA

```
role_arn='arn:aws:iam::account_id:role/role-test'
role_session_name='test'
profile_name='test'
mfa_serial='arn:aws:iam::account_id:mfa/username'
read -p "Enter your mfa token: " token
temp_role=$(aws sts assume-role \
    --role-arn $role_arn \
    --role-session-name $role_session_name
    --serial-number $mfa_serial
    --token-code $token)

export AWS_ACCESS_KEY_ID=$(echo $temp_role | jq -r
.Credentials.AccessKeyId)
export AWS_SECRET_ACCESS_KEY=$(echo $temp_role | jq -r
.Credentials.SecretAccessKey)
export AWS_SESSION_TOKEN=$(echo $temp_role | jq -r
.Credentials.SessionToken)
aws configure set aws_access_key_id $AWS_ACCESS_KEY_ID --profle
$profile_name
aws configure set aws_secret_access_key $AWS_SECRET_ACCESS_KEY --
profile $profile_name
aws configure set aws_session_token $AWS_SESSION_TOKEN --profle
$profile_name
```

- PowerShell - example
  - https://docs.aws.amazon.com/powershell/latest/reference/items/Use-STSRole.html

```
Use-STSRole
     -RoleArn <String>
     -RoleSessionName <String>
     -Policy <String>
     -DurationInSeconds <Int32>
     -ExternalId <String>
     -PolicyArn <PolicyDescriptorType[]>
     -SerialNumber <String>
     -Tag <Tag[]>
     -TokenCode <String>
     -TransitiveTagKey <String[]>
     -Select <String>
     -PassThru <SwitchParameter>
     -Force <SwitchParameter>
```

- **Usage:**
  - **$Creds =** (Use-STSRole -RoleArn "arn:aws:iam::<account:role/rolename" -RoleSessionName "MyRoleSessionName").Credentials
  - **Print:**
    - $Creds.AccessKeyId
    - $Creds.SecretAccessKey

```
 C:\Users\dev-user> Set-AWSCredential -AccessKey AKIAIYVN464EHG5XYWPQ -SecretKey 2ZJwqkpsrMGRMC0
Em7lORNM9QU
 C:\Users\dev-user> Get-S3Bucket
t-S3Bucket : Access Denied
 line:1 char:1
Get-S3Bucket
~~~~~~~~~~~~
  + CategoryInfo          : InvalidOperation: (Amazon.PowerShe...tS3BucketCmdlet:GetS3BucketCmdl
 3Bucket], InvalidOperationException
  + FullyQualifiedErrorId : Amazon.S3.AmazonS3Exception,Amazon.PowerShell.Cmdlets.S3.GetS3Bucket

 C:\Users\dev-user> $Creds = (Use-STSRole -RoleArn "arn:aws:iam::550967231773:role/my-test-role"
ame "MyRoleSessionName").Credentials
 C:\Users\dev-user> $Creds.AccessKeyId
IAYASBUYUO6GYC7R54
 C:\Users\dev-user> $Creds.SecretAccessKey
1HXY8CxWsG4/DX7+lffhUVSg8kf0533lYz5sWk
```

```
> Get-S3Bucket -Credential $Creds

BucketName
----------
cf-templates-un3alghfvsni-ap-southe
```