Using STS for cross account access in AWS :

Goal is to access resoruces from destination account using an EC2 from the source account

- Create 2 roles in both accounts . 1 in source account which will assume the 2nd in the destination account
- Both of them can be service roles for EC2 . We will change the trust relationshp later

Understanding Trust relationship

- Every role created in AWS will have a trust relationship assigned to it.
- It dictates which entity can assume this role and get the privileges (policies) associated with the role For ex -
- When we create a service role for EC2 , the trust relationship has principal as ec2.amazonaws.com
- In order to see it , create a EC2 role. Once created , besides policy there is a tab for trust relationship. Click edit trust relationship, you should be able to see a JSON as below

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- Trust relationship can also have multiple principals which can be defined in an array

```
  {
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service":[
            "ec2.amazonaws.com",
            "rds.amazonaws.com"
        ]},
      "Action": "sts:AssumeRole"
    }
  ]
  }
```

- Above role can be assumed by both ec2 and rds.
- Trust relationship can also have same or cross account roles defined as principal for it to be assumed.

Creating source account role

- Create a role for EC2 in source account.
- Copy the arn
- while creating a policy , choose 'STS' in service and in write choose 'AssumeRole'
- In the resources tab , We are supposed to give arn for the role created in destination account .
  -The policy JSON should look like this

```
 {
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": "arn:aws:iam::*DestinationAccountNumber*:role/*rolename*"
        }
    ]
 }
```

- Create a role for EC2 or any other service in destination account. Copy the arn and paste it in the policy mentioned above which is in source account
- Click on trust relationship , choose edit trust relationship
- The current principal would be for EC2 , we can remove it and replace it with arn for source account

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::*SourceAccountNumber*:role/*rolename*"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- We have now established a trust relationship between 2 accounts .
- Things to note
    1. Source role should have 'STS:AssumeROle' permission to assume destination account role arn
    2. Destination Account should have source account role arn in it's trust relationship

    3. Destination account role should have a policy attached with relevant access to its resources which source account wants access to

- Once above relationship is setup , we need to generate temperoroy credentials from source account to access resources of destination account
    1. Log in to EC2 on surce account which has the role created earlier attached to it
    2. Run below aws cli command to generate credetnials

```
aws sts assume-role --role-arn
arn:aws:iam::*DestinationAccountNumber*:role/*rolename* --role-session-name
*anysessionname*
```

3. Above command will return 3 outputs a. Access key ID b. Secret Access Key c. Session Token
4. Save the output in a notepad

- Note :- If above command returns access denied then we need to verify the assume role permissions,trust relationship and the arn of the role 5. Once we have the output , we can set them as environment variables using below commands

```
export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
export AWS_SESSION_TOKEN=AQoDYXdzEJr...<remainder of security token>
```

- Once above variables are set , we can try listing the resources from destination account depending on the policy attached to destination account role .For ex-

```
aws s3 ls --region us-east-1
```

- These credetnials are valid till the session is open

- Also the credetnials expire after a certain duration which can range from 1 hour to 12 hours .

- The duration can be seen in console when you click on the role name , just below last activity section there is 'Maximum CLI/API session duration'

  We recomend cross account access when we want to transfer data in and out of our AWS environment .for ex- Client data refresh