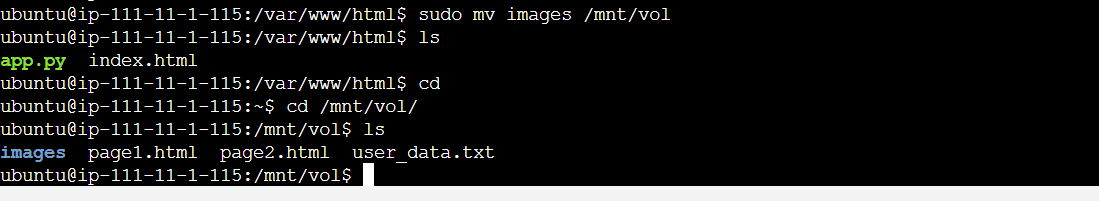
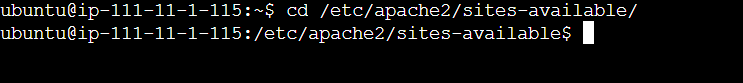


**Move all the contents from root directory to the EBS volume**

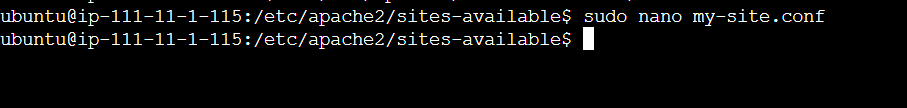


Make changes in index.html to redirect to the ebs volume

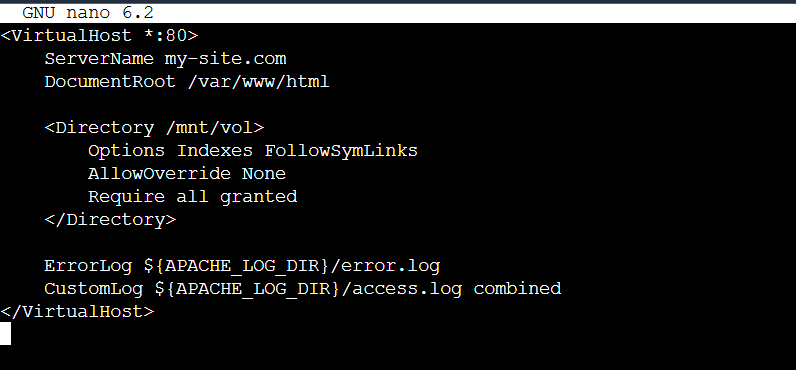
**Create a New Configuration File:**



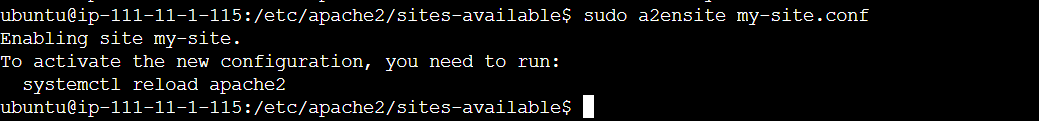
**Create a New Configuration File:**

****

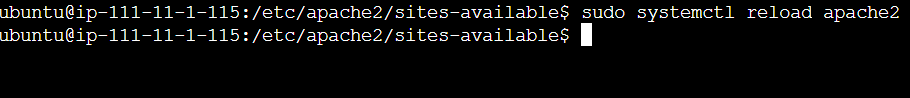
**Add Configuration for /mnt/vol**



**Enable the New Site:**



**Restart Apache:**

****

To enable cross-account access for automated snapshots and allow cross-region and cross-account data migration, you can set up IAM roles with appropriate permissions in both the source and destination accounts. Here's how you can achieve this:

1. **Create IAM Roles**: In both the source and destination AWS accounts, you need to create IAM roles. Name them as "SourceRole" and "DestinationRole" for easy identification.
2. **Sign in to the AWS Management Console**: Sign in to the AWS Management Console using your account credentials for the source AWS account.
3. **Navigate to IAM**: Go to the IAM (Identity and Access Management) dashboard by searching for "IAM" in the AWS Management Console search bar and selecting it.
4. **Create Source Role**: In the IAM dashboard, click on "Roles" in the left navigation pane, then click on the "Create role" button.
5. **Choose Trusted Entity**: Select "Another AWS account" as the trusted entity.
6. **Enter Destination Account ID**: Enter the account ID of the destination AWS account in the provided field. This allows the destination account to assume the role.
7. **Set Permissions**: Next, attach policies to the role that grant the necessary permissions for creating, copying, and deleting snapshots across regions. You can use existing policies or create custom policies tailored to your requirements.
8. **Review and Create**: Review the role configuration to ensure it's correct, then click "Next" and provide a name for the role (e.g., "SourceRole"). Optionally, you can add tags for better organization. Finally, click "Create role" to create the role.
9. **Repeat for Destination Role**: Now, repeat the above steps in the destination AWS account to create the "DestinationRole". This role should have a trust relationship with the source account, allowing it to assume the role.
10. **Verify Roles**: Once both roles are created, verify that they appear in the IAM dashboard under "Roles" in their respective AWS accounts.

By following these steps, you will have successfully created IAM roles in both the source and destination AWS accounts for cross-account access. These roles will enable the necessary permissions for automated snapshot creation and migration between the two accounts.

1. **Define Trust Relationships**: Define trust relationships for both roles to allow the source account to assume the destination role, and vice versa. Below is an example policy for the source account's SourceRole:

json

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Principal": {

"AWS": "arn:aws:iam::DESTINATION\_ACCOUNT\_ID:root"

},

"Action": "sts:AssumeRole"

}

]

}

Replace DESTINATION\_ACCOUNT\_ID with the ID of the destination AWS account. Similarly, create a trust relationship policy for the destination account's DestinationRole, allowing the source account to assume it.

1. **Attach Permissions Policies**: Attach policies to both roles that grant permissions for creating, copying, and deleting snapshots across regions. Here's an example policy that grants permissions for creating and copying snapshots:

json

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": [

"ec2:CreateSnapshot",

"ec2:CopySnapshot",

"ec2:DeleteSnapshot"

],

"Resource": "\*"

}

]

}

Attach this policy to both the SourceRole and DestinationRole.

1. **Assume Roles in Lambda Function**: Modify your Lambda function (similar to the one provided in the previous answer) to assume the destination role in the destination account. Here's a modified part of the code to assume the role:

python

import boto3

def assume\_role(account\_id, role\_name):

sts\_client = boto3.client('sts')

response = sts\_client.assume\_role(

RoleArn=f'arn:aws:iam::{account\_id}:role/{role\_name}',

RoleSessionName='CrossAccountSession'

)

return boto3.Session(

aws\_access\_key\_id=response['Credentials']['AccessKeyId'],

aws\_secret\_access\_key=response['Credentials']['SecretAccessKey'],

aws\_session\_token=response['Credentials']['SessionToken']

)

def lambda\_handler(event, context):

destination\_session = assume\_role('DESTINATION\_ACCOUNT\_ID', 'DestinationRole')

ec2 = destination\_session.client('ec2')

# Proceed with creating snapshots, copying them, etc. in the destination account

Replace 'DESTINATION\_ACCOUNT\_ID' with the ID of the destination AWS account.

1. **Testing and Monitoring**: Test your Lambda function to ensure it can successfully assume the destination role and perform the required actions in the destination account. Monitor your automated snapshot creation and migration process to ensure it's working as expected.

By following these steps, you'll enable cross-account access for automated snapshots and allow cross-region and cross-account data migration between your AWS accounts.