Transferring large data from S3 buckets across AWS accounts.

We can do that in multiple ways.

1. AWS CLI

2. AWS lambda - which we tried last time.

3. AWS SYNC

We can use aws dataSync which is 10x faster than normal transfer.

At Source Bucket:

1. Update the source bucket Policy

Sample Source bucket policy update

DIST\_ACCOUNT\_ID : New AWS account 12 digit

Role name - Role created in new aws account eg: datasysn

source\_bucket: Old Bucket name

distention\_account\_logged\_in\_user: user name created at destination account

Source Bucket Policy :

bucket --> permissions Bucket policy

{

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

"Statement": [

{

"Effect": "Allow",

"Principal": {

"AWS": [

"arn:aws:iam::DIST\_ACCOUNT\_ID:role/datasync",

"arn:aws:iam::DIST\_ACCOUNT\_ID:user/distention\_account\_logged\_in\_user"

]

},

"Action": [

"s3:GetBucketLocation",

"s3:ListBucket",

"s3:ListBucketMultipartUploads"

],

"Resource": "arn:aws:s3:::source\_bucket"

},

{

"Effect": "Allow",

"Principal": {

"AWS": [

"arn:aws:iam::DIST\_ACCOUNT\_ID:role/datasync",

"arn:aws:iam::DIST\_ACCOUNT\_ID:user/distention\_account\_logged\_in\_user"

]

},

"Action": [

"s3:AbortMultipartUpload",

"s3:DeleteObject",

"s3:GetObject",

"s3:ListMultipartUploadParts",

"s3:PutObjectTagging",

"s3:GetObjectTagging",

"s3:PutObject"

],

"Resource": "arn:aws:s3:::source\_bucket/\*"

}

]

}

2, Destination Account:

Create an IAM role

Distention account role policy

{

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

"Statement": [

{

"Action": [

"s3:GetBucketLocation",

"s3:ListBucket",

"s3:ListBucketMultipartUploads"

],

"Effect": "Allow",

"Resource": "arn:aws:s3:::source\_bucket"

},

{

"Action": [

"s3:AbortMultipartUpload",

"s3:DeleteObject",

"s3:GetObject",

"s3:ListMultipartUploadParts",

"s3:PutObject",

"s3:GetObjectTagging",

"s3:ListBucket",

"s3:PutObjectTagging"

],

"Effect": "Allow",

"Resource": "arn:aws:s3:::source\_bucket/\*"

}

]

}

Note: replace source\_bucket

3. Distention Bucket policy

{

"Version": "2008-10-17",

"Statement": [

{

"Sid": "DataSyncCreateS3LocationAndTaskAccess",

"Effect": "Allow",

"Principal": {

"AWS": [

"arn:aws:iam::DIST\_ACCOUNT\_ID:role/datasync",

"arn:aws:iam::DIST\_ACCOUNT\_ID:user/distention\_account\_logged\_in\_user"

]

},

"Action": [

"s3:GetBucketLocation",

"s3:ListBucket",

"s3:ListBucketMultipartUploads",

"s3:AbortMultipartUpload",

"s3:DeleteObject",

"s3:GetObject",

"s3:ListMultipartUploadParts",

"s3:PutObject",

"s3:GetObjectTagging",

"s3:PutObjectTagging"

],

"Resource": ["arn:aws:s3:::dist\_bucket", "arn:aws:s3:::dist\_bucket/\*"]

}

]

}

Note: update dist\_bucket

4. AWS CLI command

aws datasync create-location-s3 --s3-bucket-arn arn:aws:s3:::s3-source --s3-storage-class STANDARD --s3-config BucketAccessRoleArn="arn:aws:iam::accountnumber:role/datasync" --region us-east-1

Note: cross Check the user -- aws sts get-caller-identity

5. Login in to AWS Console --> Data sysnc

Create Task and start task

Monetoring:

We can have two ways of Monetoring

1. aws s3 sync

2. cloud watch monatoring

3. AWS Lamdba

We can have a discussion on cloud watch monatoring and AWS Lamdba

A screenshot of a computer

Description automatically generated

I need to get one s3 bucket to another account s3 bucket

**I need to migrate data from one s3 bucket to another s3 bucket in one region.**

First I need to create 2 buckets

Start bucket and destination bucket.

I need to create in the source bucket all objects.

And create a data sync task (to transfer all data)

Finally, I should get in the destination bucket all objects.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Create one region to another region s3 buckets and migrate from one bucket to another bucket using data sync**

I created 2 buckets.

1 bucket in the Virginia region and another bucket in the Ohio region

Start bucket and destination bucket.

Create a task in the start bucket.

Refresh the destination bucket you will get all objects in destination bucket!

A screenshot of a computer

Description automatically generated

This is the Virginia Start bucket.

A screenshot of a computer

Description automatically generated

This Ohio destination bucket

A screenshot of a computer screen

Description automatically generated

Wordpress mysql

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

To check iam loadbal

**How to create efs and mount it in instances:**

[**https://medium.com/geekculture/ow-to-setup-amazon-elastic-file-system-efs-and-mount-on-to-ubuntu-ec2-b47346427d5**](https://medium.com/geekculture/ow-to-setup-amazon-elastic-file-system-efs-and-mount-on-to-ubuntu-ec2-b47346427d5)

Create vpc 2 subnets in different regions and route table and internet gateway and 2 ec2 instances

And for the security group attach nfs port

And after creating efs search for networks and check security groups

Efs attach to IP address and zones

And go to the command prompt

**sudo apt-get -y install nfs-common**

* **mkdir efs**
* **ls**
* **cd efs**
* **ls**
* **and copy efs attach in that copy**
* **ls**
* **touch te.tf**
* **ls**

**needs to mount in both instances**

**Create rds and database**

**RDS:**

Create rds and save username and password

Go to the command prompt and start with

**Sudo su**

**apt-get update**

**Apt-get upgrade**

**Mysql -u root**

**Exit**

**In Ubuntu**

**Mysql -u admin12 -h (and copy endpoint) -p**

**Next**

**Show databases;**

**Create database navyaletspatchup;**

**You will get this**

**How to create rds and migrate snapshot (data) from another rds account**

Create 1 vpc,2 subnets(1a,1b), router,ig,ec2

And first give vpc enable rds

**Rds creation:**

Standard Create \_\_aurora(MySQL)\_DevTest\_\_\_names\_\_\_passwords

Cluster\_\_aurora standard

Instance configuration\_\_burstable classes----db.t3.medium

Availability & dunability—don’t create an aurora replica

Connectivy—don’t connect to an compute resources-- Ipv4

View configiration

Admin

Asdflkjh

database-443.cluster-c9csuooaqmqt.us-east-1.rds.amazonaws.com

**How to create a snapshot:**

Go to rds. database---actions---take a snapshot.

**What is docker?**

**Docker is a software platform that allows you to build, test and deploy applications quickly .docker compose is a tool that helps define, share and encapsulate multi-container applications based on yaml files that are coded in YAML files**

**Docker file:**

**base image--------docker hub**

**Do prerequisites**

**Copy code**

**Provide init**

Sudo apt-get update

Sudo apt-get install docker.io

Sudo service docker status

Sudo service docker start

Sudo su

Docker version

Docker search nginx

Docker pull nginx

Docker images

Docker ps

Docker run -d -p 8080:80 nginx

Docker ps

b2a3fbbe65b5a6321e50b8f4140b31772c5ca9d3f684ce762ab308ae04e38d21

be2989c46c1c1bab8c18031d7f8c850fa8f4243fbc049c20764c456ab482ac22

**create your own image in ecr then we create the repository and we can run that image in our local docker-machine**

**MINKUBE INSTALLATION**

Install ec2

Go to Google Minkube and install Ubuntu 22.04

<https://www.linuxbuzz.com/install-minikube-on-ubuntu/>

open mobex and sudo su , update, upgrade

docker install ubuntu 22.04

<https://docs.docker.com/engine/install/ubuntu/>

commands

sudo apt-get update

sudo apt-get install ca-certificates curl

sudo install -m 0755 -d /etc/apt/keyrings

sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc

sudo chmod a+r /etc/apt/keyrings/docker.asc

echo \

"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \

$(. /etc/os-release && echo "$VERSION\_CODENAME") stable" | \

sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update

sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

sudo docker run hello-world

docker ps

docker ps -a(we need to get HelloWorld image)

sudo usermod -aG docker $USER

newgrp docker

curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64

sudo install minikube-linux-amd64 /usr/local/bin/minikube

minikube version

curl -LO https://storage.googleapis.com/kubernetes-release/release/`curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt`/bin/linux/amd64/kubectl

chmod +x kubectl

sudo mv kubectl /usr/local/bin/

kubectl version -o yaml

minikube start --driver=docker(\*\*\*)

minikube start --driver=docker –force

minikube status

minkube get nodes(which is attached to cluster master nodes or work nodes)

kubectl describe node minikube(we can see all installation and pod cir too)

**what is their in pods?**

**The pod will have a container pod that needs to get to one particular IP address that will be within this CIDR block**

**There are 3 main elements.**

**ETCD**

**Controller**

**Scheduler**

**There are 3 namespaces**

**kube-node-lease**

**kube-system, kube-public**

kubectl get namespaces

kubectl get pods

kubectl get pods -A

kubectl get pods -n kube-node-lease

kubectl get pods -n default

kubectl create deployment nginx-web --image=nginx

kubectl get all

kubectl describe pod/nginx-web-5b757f798d-6zp5z

kubectl expose deployment nginx-web --type NodePort --port=80

kubectl get all

**service** [**url**:service](url:service) url is defined to connect with in the kubernate claster

kubectl get pods -n default

kubectl exec -it nginx-web-5b757f798d-6zp5z -n default /bin/bash

ls

cd etc/

ls

cd nginx/

ls

cd nginx/

ls

exit

kubectl get all -A (TO check deployment)

kubectl edit deployment.apps/nginx-web -n default()

**create k8 depolyment YAML file for nginx**

**create k8 depolyment YAML file for nginx along with service.YAML**

kubectl create ns traningnavya(creating namespace)

kubectl get ns(we will namespace)

ls

mkdir nginx

cd nginx/

vi deployment.yaml(we need to give namespace)

vi service.yaml

to apply this for kuburanatics

kubectl apply -f deployment.yaml

kubectl apply -f service.yaml -n traningnavya

kubectl get all -n traningnavya

kubectl edit service/nginx-traning-service -n traningnavya(to edit service loadbalancer to NodePort)

kubectl get all -n traningnavya

minikube addons list

minikube addons enable ingress

minikube addons enable dashboard

minikube addons metrics-server

minikube addons enable metrics-server

kubectl get all -A

kubectl get ns

kubectl get all -n kube-system

kubectl get ns

kubectl get all -n kubernetes-dashboard

kubectl edit service/kubernetes-dashboard -n kubernetes-dashboard

(change type:NodePort)

kubectl get all -n kubernetes-dashboard

minikube ip

kubectl get all -A

curl 192.168.49.2:32286(training port 1st)

ls

**porting:**

**public ip**

**How to install moodle:**

sudo nano /etc/php/8.0/fpm/pool.d/www.conf

1. **Save Changes**: Once you're done editing, you can save your changes by pressing **Ctrl + O** (that's the letter O, not zero). This will prompt you to confirm the filename to save to. Press **Enter** to confirm.
2. **Exit Nano**: After saving, you can exit nano by pressing **Ctrl + X**. If you've made changes and haven't saved them, nano will prompt you to save them before exiting.

https://stackoverflow.com/questions/43094726/your-password-does-not-satisfy-the-current-policy-requirements

UNINSTALL COMPONENT 'file://component\_validate\_password';