```
#!/bin/bash

# This script is intended to delete resources for EC2, ECS, and EKS across all regions. Use with extreme caution.

# Function to delete EC2 instances in all regions
delete_ec2_instances() {
    for region in $(aws ec2 describe-regions --query "Regions[].RegionName" --output text); do
        echo "Deleting EC2 instances in region $region..."
                                instances=$(aws ec2 describe-instances --region $region --query "Reservations[*].Instances[*].InstanceId" --output text --filters "Name=instance-state-name,Values=running")
        if [ -n "$instances" ]; then
            aws ec2 terminate-instances --instance-ids $instances --region $region
            echo "EC2 instances terminated in region $region"
        else
            echo "No running EC2 instances to delete in region $region"
        fi
    done
}

# Function to delete ECS clusters in all regions
delete_ecs_clusters() {
    for region in $(aws ec2 describe-regions --query "Regions[].RegionName" --output text); do
        echo "Deleting ECS clusters in region $region..."
        clusters=$(aws ecs list-clusters --region $region --query "clusterArns[]" --output text)
```

```
        for cluster in $clusters; do
                        aws ecs update-cluster-settings --cluster $cluster --region $region --settings name=containerInsights,value=disabled
            services=$(aws ecs list-services --cluster $cluster --region $region --query "serviceArns[]" --output text)
            if [ -n "$services" ]; then
                aws ecs update-service --cluster $cluster --service $services --desired-count 0 --region $region
                aws ecs delete-service --cluster $cluster --service $services --force --region $region
            fi
            aws ecs delete-cluster --cluster $cluster --region $region
            echo "ECS cluster deleted in region $region"
        done
    done
}

# Function to delete EKS clusters in all regions
delete_eks_clusters() {
    for region in $(aws ec2 describe-regions --query "Regions[].RegionName" --output text); do
        echo "Deleting EKS clusters in region $region..."
        clusters=$(aws eks list-clusters --region $region --query "clusters[]" --output text)
        for cluster in $clusters; do
            aws eks delete-cluster --name $cluster --region $region
            echo "EKS cluster $cluster deleted in region $region"
        done
    done
```

```
}

# Function to delete ECR repositories in all regions
delete_ecr_repositories() {
    for region in $(aws ec2 describe-regions --query "Regions[].RegionName" --output text); do
        echo "Deleting ECR repositories in region $region..."
                                repositories=$(aws ecr describe-repositories --region $region --query "repositories[].repositoryName" --output text)
        for repo in $repositories; do
            images=$(aws ecr list-images --repository-name $repo --region $region --query 'imageIds[*]' --output text)
            if [ -n "$images" ]; then
                    aws ecr batch-delete-image --repository-name $repo --image-ids imageTag=$images --region $region
                echo "Images deleted from $repo in region $region"
            fi
            aws ecr delete-repository --repository-name $repo --region $region --force
            echo "ECR repository $repo deleted in region $region"
        done
    done
}


# Function to delete Auto Scaling groups in all regions
delete_auto_scaling_groups() {
    for region in $(aws ec2 describe-regions --query "Regions[].RegionName" --output text); do
```

```
        echo "Deleting Auto Scaling groups in region $region..."
                asgs=$(aws autoscaling describe-auto-scaling-groups --region $region --query "AutoScalingGroups[].AutoScalingGroupName" --output text)
        for asg in $asgs; do
            aws autoscaling update-auto-scaling-group --auto-scaling-group-name $asg --min-size 0 --max-size 0 --desired-capacity 0 --region $region
            aws autoscaling delete-auto-scaling-group --auto-scaling-group-name $asg --force-delete --region $region
            echo "Auto Scaling group $asg deleted in region $region"
        done
    done
}

# Function to delete Load Balancers (ELB and ALB) in all regions
delete_load_balancers() {
    for region in $(aws ec2 describe-regions --query "Regions[].RegionName" --output text); do
        echo "Deleting Load Balancers in region $region..."
        # Classic Load Balancers
                clbs=$(aws elb describe-load-balancers --region $region --query "LoadBalancerDescriptions[].LoadBalancerName" --output text)
        for clb in $clbs; do
            aws elb delete-load-balancer --load-balancer-name $clb --region $region
            echo "Classic Load Balancer $clb deleted in region $region"
        done
        # Application and Network Load Balancers
                anlbs=$(aws elbv2 describe-load-balancers --region $region --query
```

```
"LoadBalancers[].LoadBalancerArn" --output text)
        for anlb in $anlbs; do
            aws elbv2 delete-load-balancer --load-balancer-arn $anlb --region $region
            echo "Application/Network Load Balancer $anlb deleted in region $region"
        done
    done
}



######### ------ Delete all services
delete_ec2_volumes() {
    for region in $(aws ec2 describe-regions --query "Regions[].RegionName" --output text); do
        echo "Deleting EC2 volumes in region $region..."
         volumes=$(aws ec2 describe-volumes --region $region --query "Volumes[].VolumeId" --output
text)
        for volume in $volumes; do
            aws ec2 delete-volume --volume-id $volume --region $region
            echo "Volume $volume deleted in region $region"
        done
    done
}

delete_ec2_snapshots() {
    for region in $(aws ec2 describe-regions --query "Regions[].RegionName" --output text); do
        echo "Deleting EC2 snapshots in region $region..."
            snapshots=$(aws ec2 describe-snapshots --owner-ids $(aws sts get-caller-identity --query
```

```
"Account" --output text) --region $region --query "Snapshots[].SnapshotId" --output text)
        for snapshot in $snapshots; do
            aws ec2 delete-snapshot --snapshot-id $snapshot --region $region
            echo "Snapshot $snapshot deleted in region $region"
        done
    done
}

delete_ec2_key_pairs() {
    for region in $(aws ec2 describe-regions --query "Regions[].RegionName" --output text); do
        echo "Deleting EC2 key pairs in region $region..."
            key_pairs=$(aws ec2 describe-key-pairs --region $region --query "KeyPairs[].KeyName"
--output text)
        for key_pair in $key_pairs; do
            aws ec2 delete-key-pair --key-name $key_pair --region $region
            echo "Key pair $key_pair deleted in region $region"
        done
    done
}

delete_ec2_security_groups() {
    for region in $(aws ec2 describe-regions --query "Regions[].RegionName" --output text); do
        echo "Deleting EC2 security groups in region $region..."
        # Exclude default security group
                security_groups=$(aws ec2 describe-security-groups --region $region --query
"SecurityGroups[?GroupName != 'default'].GroupId" --output text)
```

```
        for sg in $security_groups; do
            aws ec2 delete-security-group --group-id $sg --region $region
            echo "Security group $sg deleted in region $region"
        done
    done
}

delete_ec2_amis() {
    for region in $(aws ec2 describe-regions --query "Regions[].RegionName" --output text); do
        echo "Deleting EC2 AMIs in region $region..."
            amis=$(aws ec2 describe-images --owners self --region $region --query "Images[].ImageId"
--output text)
        for ami in $amis; do
            aws ec2 deregister-image --image-id $ami --region $region
            echo "AMI $ami deregistered in region $region"
        done
    done
}

# Make sure to call these functions in your script




# Call functions
```

```
delete_ec2_instances

delete_ecs_clusters

delete_eks_clusters

delete_ecr_repositories

delete_auto_scaling_groups

delete_load_balancers

delete_ec2_volumes

delete_ec2_snapshots

delete_ec2_key_pairs

delete_ec2_security_groups

delete_ec2_amis



echo "Deletion script completed."
```