Inventory data:

aws ec2 describe-instances --query 'Reservations[].Instances[].{Name:Tags[?Key==`Name`]|[0].Value,Instance:InstanceId,Subnet:SubnetId,PublicIP:PublicIpAddress,PrivateIP:PrivateIpAddress}' --output table --profile Indigo

Modifiied

aws ec2 describe-instances --query 'Reservations[].Instances[].{Name:Tags[?Key==`Name`]|[0].Value,Instance:InstanceId,Subnet:SubnetId,PublicIP:PublicIpAddress,PrivateIP:PrivateIpAddress,AZ:Placement.AvailabilityZone}' --output table --profile Indigo

Tags

aws ec2 describe-instances --query 'Reservations[].Instances[].[InstanceId,InstanceType,Platform,State.Name,PrivateIpAddress,Tags[?Key==`Name`].Value | [0],Tags[?Key==`Application`].Value | [0],Tags[?Key==`Application Owner`].Value |[0],Tags[?Key==`Environment`].Value | [0]]' --output table --profile Indigo

**Latest**

aws ec2 describe-instances --query 'Reservations[].Instances[].[InstanceId,InstanceType,Platform,State.Name,VpcId,SubnetId,PublicIpAddress,PrivateIpAddress,Placement.AvailabilityZone,Tags[?Key==`Name`].Value | [0],Tags[?Key==`Application`].Value | [0],Tags[?Key==`Application Owner`].Value |[0],Tags[?Key==`Environment`].Value | [0]]' --output text --profile Indigo

Filter:

aws ec2 describe-instances --filters Name=tag:Application,Values=Inform Name=tag:Environment,Values=Production --region ap-south-1 --profile Indigo --query 'Reservations[].Instances[].{Name:Tags[?Key==`Name`]|[0].Value,Instance:InstanceId}' --output table

Create Tag

aws ec2 --region ap-southeast-1 --profile Indigo create-tags --resources vol-0c02c02aa0ce077f4 --tags Key=Application,Value='Inform' Key='Application Owner',Value='Meenakshi Jain' Key=Environment,Value='Production'

**Get Volume information attached to an ec2**

aws ec2 describe-volumes --output text --query 'Volumes[\*].Attachments[].{VolumeID:VolumeId,InstanceID:InstanceId}' --profile Indigo

**Query volumes which doesn’t have the mentioned tag:**

aws ec2 describe-volumes --query 'Volumes[?!not\_null(Tags[?Key == `Application`].Value)] | [].[VolumeId,Size]' --profile Indigo --output text

aws ec2 describe-volumes --query 'Volumes[\*].[VolumeId,State,Size]' --output table --profile Indigo

**Get unused volumes:**

aws ec2 describe-volumes --filter "Name=status,Values=available" --query 'Volumes[\*].[VolumeId,State,Size]' --output table --profile Indigo

aws ec2 describe-volumes --query 'Volumes[\*].[VolumeId,State,Size]' --output table --profile Indigo

aws ec2 describe-volumes --output text --query 'Volumes[\*].Attachments[].{VolumeID:VolumeId,Size,InstanceID:InstanceId}' --profile Indigo

**RDS DETAILS**

**aws rds describe-db-instances --query "DBInstances[\*].[DbiResourceId,DBInstanceIdentifier,Endpoinroot@rohit:/home/rohit/Downloads# aws rds describe-db-instances --query "DBInstances[\*].[DbiResourceId,DBInstanceIdentifier,Endpoint.Address,DBInstanceClass,Engine,EngineVersion,AvailabilityZone,AllocatedStorage]" --output text --region ap-south-1 --profile Indigoprofile Indigo**

**aws rds describe-db-instances --query "DBInstances[\*].[DbiResourceId,DBInstanceIdentifier,Endpoint.Address,DBInstanceClass,Engine,EngineVersion,DBSubnetGroup.VpcId,AvailabilityZone,AllocatedStorage]" --output text --region ap-south-1 --profile Indigo**

**VPC**

**aws ec2 describe-vpcs --query "Vpcs[\*].[VpcId,CidrBlock,Tags]" --region ap-south-1 --profile Indigo**

**Elastic IP**

**aws ec2 describe-addresses --region ap-south-1 --query 'Addresses[\*].[AllocationId,PublicIp,PrivateIpAddress,NetworkBorderGroup,InstanceId,Tags[?Key==`Name`].Value | [0]]' --profile Indigo --output text**

**OS INFORMATION : CONDITION REQUIRE SSM AGENT ON MACHINE**

**aws ssm describe-instance-information --query 'InstanceInformationList[\*].[InstanceId,PlatformType,PlatformName]' --region ap-southeast-1 --output text --profile Indigo | sort**

**Volumes Information:**

**aws ec2 describe-volumes \**

**--query 'Volumes[\*].{ID:VolumeId,InstanceId:Attachments[0].InstanceId,AZ:AvailabilityZone,Size:Size}'**

**aws ec2 describe-volumes --query 'Volumes[\*].{ID:VolumeId,InstanceId:Attachments[0].InstanceId,AZ:AvailabilityZone,Size:Size}' --profile Indigo --region ap-south-1 --output text**

**aws ec2 describe-volumes --query 'Volumes[\*].{ID:VolumeId,Enryption:Encrypted,InstanceId:Attachments[0].InstanceId,AZ:AvailabilityZone,Size:Size}' --profile Indigo --region eu-central-1 --output text**

**aws s3api list-objects-v2 --bucket groupbooking --query "Contents[?(LastModified>='2020-09-20T15:00:00.00GMT+0530' && LastModified<='2020-09-20T19:00:00.00GMT+0530')].Key" --profile Indigo**

**aws rds describe-db-instances --query "DBInstances[\*].[DbiResourceId,DBInstanceIdentifier,Endpoint.Address,DBInstanceClass,Engine,EngineVersion,DBSubnetGroup.VpcId,AvailabilityZone,AllocatedStorage,**Tags[?Key==`Name`].Value | [0],Tags[?Key==`Application`].Value | [0],Tags[?Key==`Application Owner`].Value |[0],Tags[?Key==`Environment`].Value | [0]]’ **--output text --region ap-south-1 --profile Indigo**

**List cloudwatch alarms**

**aws cloudwatch describe-alarms --query MetricAlarms[\*].[AlarmName] --region eu-central-1 --profile Indigo --output text**

**FOR MEMORY DATA WEEKLY OR MONTHLY**

**for i in $(cat abc.txt)**

**do**

**aws cloudwatch get-metric-statistics --namespace CWAgent --metric-name mem\_used\_percent --period 3600 --statistics Maximum --dimensions Name=InstanceId,Value=$i --start-time 2020-11-26T13:05:00 --end-time 2020-12-02T23:10:00 --profile Indigo --output text | sort -n | awk '{print $2}' | awk '{ total += $1; count++ } END { print total/count }'**

**done**

**FOR CPU DATA WEEKLY OR MONTHLY**

**for i in $(cat abc.txt)**

**do**

**aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization --period 3600 --statistics Maximum --dimensions Name=InstanceId,Value=$i --start-time 2020-11-26T13:05:00 --end-time 2020-12-02T23:10:00 --profile Indigo --output text | sort -n | awk '{print $2}' | awk '{ total += $1; count++ } END { print total/count }'**

**done**