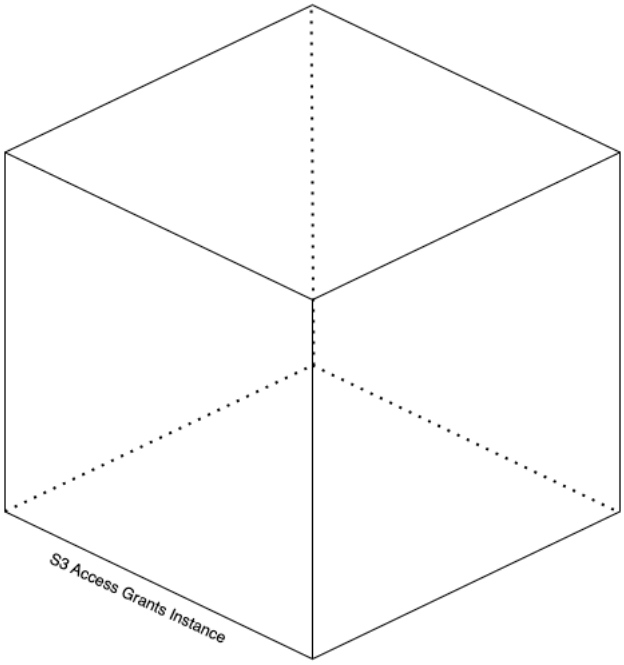
Configuring Amazon S3 security settings and access controls

- Getting started at an AWS hosted workshop
- S3 Security Best Practices
  - Prepare Your Lab
    - Attach IAM Role to EC2 Instance
    - Connect to the EC2 Instance
    - Bucket Name
  - Lab 1 - S3 Security Exercises
    - Require HTTPS
    - Require SSE-KMS Encryption
    - Restrict Access to an S3 VPC Endpoint
    - Use AWS Config Rules to Detect a Public Bucket
    - Use Amazon Access Analyzer for S3
  - Lab 2 - S3 Access Grants
    - **S3 Access Grants Lab - Initial Setup**
    - Configure S3 Access Grants for IAM user
  - Lab 3 - Enabling Malware Protection for S3 by using GuardDuty
  - Lab 4 - S3 Access Control Lists
  - Lab Summary

S3 Access Grants Lab - Initial Setup

S3 Access Grants Instance

This is the first step to start using Amazon S3 Access Grants.
The S3 Access Grants instance serves as the container for your S3 Access Grants resources, which include registered locations and grants. You can create only one S3 Access Grants instance per AWS Region per account.



CLI | Web Console

Using the CLI

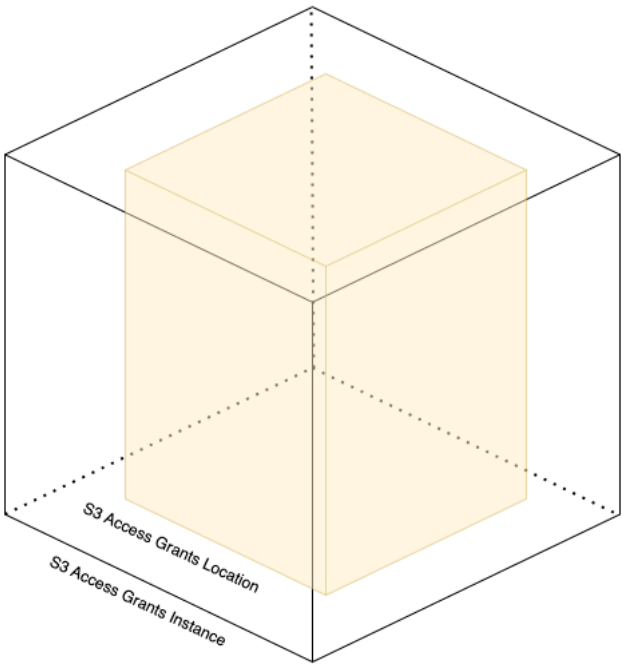
Open the Cloud Shell and copy/paste the command below in the shell.

```
AWS_ACCOUNT_ID=$(aws sts get-caller-identity | jq -r .Account)
echo "export AWS_ACCOUNT_ID=$AWS_ACCOUNT_ID" > workshop.env
source workshop.env

# Create an S3 Access Grants Instance.
aws s3control create-access-grants-instance \
--account-id $(aws sts get-caller-identity | jq -r .Account) \
--region $AWS_DEFAULT_REGION

# Fetch the S3 Access Grants Instance ARN
S3AG_INSTANCE_ARN=$(aws s3control get-access-grants-instance --account-id $AWS_ACCOUNT_II
echo "export S3AG_INSTANCE_ARN=$S3AG_INSTANCE_ARN" >> workshop.env
```

S3 Access Grants Location

After you create an Amazon S3 Access Grants instance in a AWS Region in your account, you can register an S3 location in that instance. A location is an S3 resource that contains data that you want to grant access to. You can register the default location, s3://, which is all of your buckets in the AWS

Region, and then narrow the scope of access later, when you create individual access grants. You can also register a specific bucket or a bucket and prefix as a location.



For S3 Access Grants to vend credentials to users, it must have an IAM Role with the minimum permissions required to work.

First, the IAM role must have a trust relationship policy to allow the service `access-grants.s3.amazonaws.com` to execute some API actions. The required API actions are:

- `sts:AssumeRole`
- `sts:SetSourceIdentity`

Here is an example of trust relationship policy:

CLI | **Example**

Copy and paste the command below in your Cloud Shell to create the file `trust-policy.json`

```
cat << 'EOF' > trust-policy.json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3AccessGrantsTrust",
            "Effect": "Allow",
            "Principal": {
                "Service": "access-grants.s3.amazonaws.com"
            },
            "Action": [
                "sts:AssumeRole",
                "sts:SetSourceIdentity",
                "sts:SetContext"
            ]
        }
    ]
}
EOF
```

The action `sts:SetContext` included in the policy above is only required for the Module 2. It allows S3 Access Grants location to add directory user context to the credentials generated.

Next, you need to create an IAM policy with the permissions that S3 Access Grants service will be able to generate credentials on your behalf. This policy has been crafted to scope down the permissions to

only the Access Grants instance and account that you are using today. You can use this policy as an example to your own needs.

CLI | **Example**

Copy and paste the command below in your Cloud Shell to generate the file `iam-policy.json`

```
source workshop.env

cat << EOF > iam-policy.json
{
   "Version":"2012-10-17",
   "Statement": [
       {
         "Sid": "ObjectLevelReadPermissions",
         "Effect":"Allow",
         "Action":[
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:GetObjectAcl",
            "s3:GetObjectVersionAcl",
            "s3:ListMultipartUploadParts"
         ],
         "Resource":[
            "arn:aws:s3:::*"
         ],
         "Condition":{
            "StringEquals": { "aws:ResourceAccount": "$AWS_ACCOUNT_ID" },
            "ArnEquals": {
                "s3:AccessGrantsInstanceArn": ["$S3AG_INSTANCE_ARN"]
            }
          }
      },
      {
         "Sid": "ObjectLevelWritePermissions",
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:PutObjectAcl",
            "s3:PutObjectVersionAcl",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion",
            "s3:AbortMultipartUpload"
         ],
         "Resource":[
            "arn:aws:s3:::*"
         ],
         "Condition":{
            "StringEquals": { "aws:ResourceAccount": "$AWS_ACCOUNT_ID" },
            "ArnEquals": {
                "s3:AccessGrantsInstanceArn": ["$S3AG_INSTANCE_ARN"]
            }
         }
      },
      {
         "Sid": "BucketLevelReadPermissions",
         "Effect":"Allow",
         "Action":[
            "s3:ListBucket"
         ],
         "Resource":[
            "arn:aws:s3:::*"
         ],
         "Condition":{
            "StringEquals": { "aws:ResourceAccount": "$AWS_ACCOUNT_ID" },
            "ArnEquals": {
                "s3:AccessGrantsInstanceArn": ["$S3AG_INSTANCE_ARN"]
            }
         }
      },
      {
         "Sid": "KMSPermissions",
         "Effect":"Allow",
         "Action":[
            "kms:Decrypt",
```

```
            "kms:GenerateDataKey"
         ],
         "Resource":[
            "*"
         ]
      }
   ]
}
EOF
```

We have highlighted the lines that you need to change before use.
Change the `AWS Region`, `accountId` and `instancdId` with the current values of your environment.

CLI | Web Console

Using the CLI

Open the Cloud Shell and copy/paste the command below in the shell.

You can register the default location, `s3://`, or a custom location in your S3 Access Grants instance. Make sure that you have created the two policies above in your Cloud Shell or you have uploaded the two files to your Cloud Shell Run the following command:

```
aws iam create-role --role-name accessGrantsLocationRole \
 --region $AWS_DEFAULT_REGION \
 --assume-role-policy-document file://trust-policy.json
```

Next, you will attach the IAM policy to the IAM role that you just created above.

```
aws iam put-role-policy \
--role-name accessGrantsLocationRole \
--policy-name accessGrantsLocationRole \
--policy-document file://iam-policy.json
```

Now you will create the S3 Access Grant Location.

```
source workshop.env
aws s3control create-access-grants-location \
 --account-id $AWS_ACCOUNT_ID \
 --location-scope s3:// \
 --iam-role-arn arn:aws:iam::$AWS_ACCOUNT_ID:role/accessGrantsLocationRole
```

Step complete
You have completed the initial setup. Now, we will configure the grants that will allow users to access the data.

Previous Next