

# Encryption enforcement

*Enforcing encryption of AWS resources for Data Protection*

4 minute read Published: 5 May, 2021

## Data Protection Enforcement #

The MPS environment was examined by AWS Professional Services and they suggested several improvements based on the VWFS IT Standards (E-OHB - G10.P01 - IT Compliance) and the Cloud Security Alliance - Cloud Control Matrix (CSA CCM).

Those improvements were thoroughly assessed and tested in collaboration with GRC.

The Data Protection enforcements take effect on all new MPS tenants provisioned after March 7, 2021. Existing tenants are not affected as those are breaking changes.

CloudGuard is used as an organizational measurement and compensatory control for existing tenants.

## Affected AWS resources and impact #

The enforcements are mainly encryption related and affect the following AWS services

- EC2
- S3
- RDS
- EFS

If you try to run one of the mentioned actions below via UI, CLI or API and you do not adhere to the conditions, you will get an `access denied` error.

MPS enforces the restrictions via Service Control Policies (SCP) which are policies that can be used to manage permissions at the AWS Organization level.

Please keep in mind that those SCPs snippets below are simplified. The comprehensive SCP can be found [here](#).

## EC2 restrictions #

MPS enforces, that EC2 instances and EBS volumes can only be created if they are encrypted. Therefore we restrict the following:

- EC2 instances can only be created if the root volume is encrypted
- EBS volumes and EBS snapshots can only be created if they are encrypted

## Affected actions #

```
"Effect": "Deny",
"Action": [
    "ec2:CreateSnapshots",
    "ec2:CreateSnapshot",
    "ec2:RunInstances",
    "ec2:CreateVolume"
],
"Resource": "*",
"Condition": { "Bool": { "ec2:Encrypted": "false" } }
```

## S3 restrictions #

MPS enforces that all objects in S3 buckets are encrypted by default and insecure transport for S3 is not possible. Therefore we restrict the following:

- Objects can only be uploaded to S3 if the header `s3:x-amz-server-side-encryption` is present
- S3 actions can only be executed if secure transport is enabled via bucket policy

## Affected actions #

```
"Effect": "Deny",
"Action": "s3:PutObject",
"Resource": "*",
"Condition": {
  "StringNotEquals": { "s3:x-amz-server-side-encryption": [ "aws:kms", "AES256"]},
  "Null": { "s3:x-amz-server-side-encryption": true }
```


```
"Effect": "Deny",
"Action": "s3:*",
"Condition": { "Bool": { "aws:SecureTransport": "false" } }
```

## Examples #

### Upload object via AWS Console

If you want to upload objects directly via AWS console, you need to select specify an encryption key and use default encryption bucket settings .

#### Server-side encryption settings

Server-side encryption protects data at rest. [Learn more](#) 

**Server-side encryption**

Do not specify an encryption key

Specify an encryption key

**Encryption settings**

Use default encryption bucket settings

Override default encryption bucket settings

**Encryption key type**

To upload an object with a customer-provided encryption key (SSE-C), use the AWS CLI, AWS SDK, or Amazon S3 REST API.

AWS KMS managed customer master key (SSE-KMS)

**Encryption key ARN**

arn:aws:kms:eu-central-1:320652103645:alias/aws/s3

**Bucket Key**

Enabled

### Upload object via CLI

You need to include `--server-side-encryption=AES256` or `--server-side-encryption=aws:kms` in your `aws s3api put-object` call.

Read more about it [here](#)

```
aws s3api put-object \  
  --bucket=<<your-bucket>> \  
  --key=<<your-key>> \  
  --body=<<body>> \  
  --server-side-encryption=<<AES256 || aws:kms>>
```

## Upload Object via SDK

You need to include `ServerSideEncryption: 'AES256'` or `ServerSideEncryption: 'aws:kms'` in your `s3.putObject()` call. The following snippet uses NodeJS as an example which can be translated to other languages.

```
s3.putObject({  
  Bucket: <<your-bucket>>,  
  Key: <<your-key>>,  
  Body: <<body>>,  
  ServerSideEncryption: <<AES256 || aws:kms>>,  
})
```

## Deployment via Serverless

Serverless supports out of the box the configuration of an existing deployment bucket and specifying the encryption method.

We advise our customers to leverage the MPS provided compliant [deployment buckets](#) which already fulfill all necessary compliance requirements.

Afterwards you would only need to use the `serverSideEncryption` parameter with `serverSideEncryption: AES256` or `serverSideEncryption: aws:kms`.

```
provider:  
  deploymentBucket:  
    name: <<your-custom-deployment-bucket>>  
    serverSideEncryption: AES256 || aws:kms
```

## Using S3 as terraform state

If you are using S3 for storing your terraform state, then you need to add `encrypt = true` to you state definition like below.

```

terraform {
  required_version = ">= 0.14"

  backend "s3" {
    bucket = "<<your-bucket>>"
    region = "<<your-region>>"
    profile = "<<your-profile>>"
    key     = "<<your-key>>.tfstate"

    dynamodb_table = "terraform_lock"
    encrypt          = true
  }
}

```

## RDS restrictions #

MPS enforces that RDS cluster creation, Aurora instance creation and cluster restore from S3 is only possible with encryption enabled. Therefore we restrict the following:

- RDS Aurora instance can only be created if it is encrypted by default
- RDS cluster can only be created if it is encrypted by default
- RDS cluster can only be restored from S3 if it is encrypted by default

## RDS action restrictions #

```

"Effect": "Deny",
"Action": [
  "rds:RestoreDBClusterFromS3",
  "rds:CreateDBCluster"
],
"Resource": "*",
"Condition": { "Bool": { "rds:StorageEncrypted": "false" } }

```

```

"Effect": "Deny",
"Action": "rds:CreateDBInstance",
"Resource": "*",
"Condition": {
  "StringNotLike": { "rds:DatabaseEngine": [ "aurora*", "docdb" ] },
  "Bool": { "rds:StorageEncrypted": "false" }
}

```

## EFS restrictions #

MPS enforces that EFS creation is only possible with encryption enabled.

## EFS action restrictions #

```
"Effect": "Deny",  
"Action": "elasticfilesystem:CreateFileSystem",  
"Resource": "*",  
"Condition": { "Bool": { "elasticfilesystem:Encrypted": "false" } }
```

**Get in touch**

[Go to Support page](#)

Published by Crispin Weissfuss 5 May, 2021 using 709 words.

---

MPS Team

Made with love