VM MIGRATION TO AWS

Before attempting to import a VM, you may need to prepare your AWS environment by creating a service account with appropriate permissions, and you must prepare your locally hosted VM so that it will be accessible once it is imported into AWS.

⚠IMPORTANT⚠

- It is strongly recommended that you import VMs as Amazon Machine Images (AMI) instead of instances.
- All yellow-highlighted portions of this document are subject to change depending on your **OWN** system configurations. **Note:** No highlighted parts here are meant to be used as a default.

CONVERT VM TO AN IMAGE

VM migration supports the following image format for importing VMs

- Open Virtual Appliance (OVA) image format
- Stream-optimized ESX Virtual Machine Disk (VMDK) image format
- Fixed and Dynamic Virtual Hard Disk (VHD/VHDX) image format

Before converting a virtual machine to an image, ensure it is **"powered off"** in your hypervisor – not suspended. If it is suspended, launch the virtual machine, and shut it down.

1a. Select the virtual machine and Export Appliance from the Settings tab.

1b. Provide an exportation location.

1c. After export, this VM becomes an image file

UPLOAD THE IMAGE TO AMAZON S3

2a. Create an S3 in the AWS console of unique name. **Note. This name cannot be changed later.**

2b. Upload your VM image file to your Amazon S3 bucket using the upload tool of your choice.

CREATE SERVICE ROLE

VM migration requires a role to perform certain operations on your behalf. You **MUST** create a service role named ***vmimport*** with a trust relationship policy document that allows VM migration to assume the role, and you must attach an IAM policy to the role.

3a. In the AWS console, identify the Identity and Access Management (IAM) service.
3b. Create a service role in the dashboard
3c. Choose the EC2 service
3d. In "Attach permissions policies" choose "Create policy"

i. Creating Policy

Add the following policy to the JSON file:

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect": "Allow",
         "Action": [
            "s3:GetBucketLocation",
            "s3:GetObject",
            "s3:ListBucket"
         ],
         "Resource": [
            "arn:aws:s3:::disk-image-file-bucket",#S3 bucket name
            "arn:aws:s3:::disk-image-file-bucket/*"
         ]
      },
      {
         "Effect": "Allow",
         "Action": [
            "s3:GetBucketLocation",
            "s3:GetObject",
            "s3:ListBucket",
            "s3:PutObject",
            "s3:GetBucketAcl"
         ],
         "Resource": [
            "arn:aws:s3:::export-bucket",
            "arn:aws:s3:::export-bucket/*"
         ]
      },
      {
         "Effect": "Allow",
         "Action": [
            "ec2:ModifySnapshotAttribute",
            "ec2:CopySnapshot",
```

```
            "ec2:RegisterImage",
            "ec2:Describe*"
        ],
        "Resource": "*"
    }
  ]
}
```

Review the policy, name the policy, and create the policy.

3e. Back in the role, in "*Attach Permissions Policies*", attach the newly created policy with appropriate name.

3f. Name the role ***vmimport*** and validate the role creation

Required Permissions for IAM Users

If you have not created a user for the VM migration yet, go ahead and create one and give the user an AWS access type, at least the programmatic access.

4a. In the permissions, you want to go to "*Attach existing policies directly*".

4b. Give the user "***AdministratorAccess***" permission in "*Set Permissions*" and a **new policy**

i. Creating policy

Add the following policy to the JSON file:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListAllMyBuckets"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:CreateBucket",
        "s3:DeleteBucket",
        "s3:DeleteObject",
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:PutObject"
```

```
        ],
        "Resource": ["arn:aws:s3::: disk-image-file-bucket
","arn:aws:s3::: disk-image-file-bucket /*"]
      },
      {
        "Effect": "Allow",
        "Action": [
          "ec2:CancelConversionTask",
          "ec2:CancelExportTask",
          "ec2:CreateImage",
          "ec2:CreateInstanceExportTask",
          "ec2:CreateTags",
          "ec2:DeleteTags",
          "ec2:DescribeConversionTasks",
          "ec2:DescribeExportTasks",
          "ec2:DescribeExportImageTasks",
          "ec2:DescribeImages",
          "ec2:DescribeInstanceAttribute",
          "ec2:DescribeInstanceStatus",
          "ec2:DescribeInstances",
          "ec2:DescribeSnapshots",
          "ec2:DescribeTags",
          "ec2:ExportImage",
          "ec2:ImportInstance",
          "ec2:ImportVolume",
          "ec2:StartInstances",
          "ec2:StopInstances",
          "ec2:TerminateInstances",
          "ec2:ImportImage",
          "ec2:ImportSnapshot",
          "ec2:DescribeImportImageTasks",
          "ec2:DescribeImportSnapshotTasks",
          "ec2:CancelImportTask"
        ],
        "Resource": "*"
      }
    ]
}
```

Review the policy, name the policy, and create policy

4c. Back in the user, in "*Set permissions*", attach the newly created policy.

4d. Review and validate the user and set permissions

4e. After creating the user, **download and save user credentials file** containing the secret key and the access key.

NOTE:

If you are logged in as an AWS Identity and Access Management (IAM) user, you will need an "***AdministratorAccess***" permission if not granted one and the new policy above in your IAM policy to use for VM migration

CREATE CONTAINERS.JSON FILE

When migrating your VM to AWS, the image is exported via a container feature into your AWS workspace. A container file is a JSON file (a file with the extension .json) that contains certain or all tags, triggers, and variables you want to migrate to your AWS.

Note. It is preferrable to create this file in a medium with which the extension can be easily changed when saved as .json file such as Notepad for example. On your computer, create a file called "***containers.json***"

In this file, you want to have the following piece of code:

```
[
  {
    "Description": "My Server",
    "Format": "ova" or "vmdk",
    "UserBucket": {
        "S3Bucket": " disk-image-file-bucket ",
        "S3Key": "my-server-vm.ova" or "my-server-vm.vmdk"
    }
}]
```

Save the file as a .json and **note the path** of the file on the used device.

Install and Configure AWS Command Line Interface (CLI)

If you have had access to AWS CLI before, skip to **step 5a.**

Do not have AWS CLI? No worries. Use the link below to install AWS CLI on your device for the preferred OS. https://aws.amazon.com/cli/

Download and install AWS CLI on your computer

- Launch your command prompt and to check which version you have installed by running the ***aws –version*** command
- The returned value provides the current version you have installed. The following example shows that the version running is 2.0.47.

```
$ aws --version
aws-cli/2.0.47  Python/3.7.4  Linux/4.14.133-113.105.amzn2.x86_64
botocore/2.0.0
```

5a. Run the command **`$aws configure`** in the command prompt.
5b. It will prompt you to switch to an IAM user using its credentials.
5c. In the downloaded and saved file from **step 4e.,** acquire the credentials and input them in the command prompt as follows:

```
AWS Access Key ID [None]: Access key from file
AWS Secret Access Key [None]: Secret key from file
Default region name [None]: S3 bucket region (Example: us-east-1)
Default output format [None]: json
```

These parameters should give you access as the IAM User created in **step 4d.**
To know if you successfully accessed the console, run the command **`$aws s3 ls`**. This command should list all the S3 buckets in your workspace including the one created in **step 2a.**

5d. Use the following command in the command prompt to import your image

```
aws ec2 import-image --description "My server VM" --disk-containers "file://C:\import\containers.json"
```

At this point, your VM is successfully being migrated to your AWS console as an Amazon Machine Image (AMI).

Note. You can monitor the status of your task by running the following command:

```
aws ec2 describe-import-image-tasks --import-task-ids import-ami-1234567890abcdef
```

- If the status value says "**`active`**", then the migration task is in progress and you can check the percentage completion as well
- If the status value says "**`completed`**", then the import task is completed, and the AMI is ready to use.

Launching a migrated VM as an EC2 instance

- Go to EC2 service
  - Launch instance
    - **Step 1:** "Choose AMI"
      - Under "Quick Start"
        - Choose "My AMI".
          - Choose your AMI imported and use it to launch your instance.

HAPPY MIGRATION 😎