Standard Operating Procedure (SOP) for Creating an AWS Virtual Environment

Objective:

To import an on-premises virtual machine to AWS and create an AWS virtual environment.

Prerequisites:

1. Access to the on-premises virtual machine.
2. AWS account with necessary permissions.
3. AWS CLI installed.
4. Virtual machine exported as an OVA file.
5. An S3 bucket for storing the OVA file.

Procedure:

Step 1: Prepare the Virtual Machine

1. **Check VM Configuration:**
    - Ensure the virtual machine (VM) is in a fully powered off state.
    - Confirm there are no VM-specific software like VMware tools installed.
    - Ensure you can connect via RDP when the VM is running on AWS.
2. **Export VM to OVA File:**
    - Open VMware Workstation (or any compatible virtualization product).
    - Select the VM and ensure it is powered off.
    - Go to `File` > `Export to OVF`.
    - Change the default file extension from `.ovf` to `.ova`.
    - Name the file and save it to a desired location.

Step 2: Install AWS CLI

1. **Download and Install:**
    - Navigate to the AWS CLI download page.
    - Download the appropriate version for your operating system (Windows, Linux, MacOS).
    - Follow the installation instructions provided on the webpage.
2. **Verify Installation:**

- Open a command prompt.
- Run the command `aws --version` to verify the installation.

Step 3: Configure AWS CLI

1. **Open Command Prompt:**
   - Type `aws configure` and press Enter.
2. **Enter Configuration Details:**
   - **AWS Access Key ID:** Obtain from the AWS Management Console under IAM > Users > Security Credentials.
   - **AWS Secret Access Key:** Same as above.
   - **Default Region Name:** Choose the region code (e.g., `eu-west-2` for London).
   - **Default Output Format:** Use `json`.

Step 4: Create IAM Role for VM Import

1. **Create JSON File:**
   - Use the JSON code provided on the AWS documentation for VM import roles.
   - Save the JSON code in a file, for example, `trust-policy.json`.
2. **Create IAM Role:**
   - Open the command prompt.

Execute the command to create the role using the JSON file:
bash
Copy code

```
aws iam create-role --role-name vmimport --assume-role-policy-document file://trust-policy.json
```

   -
3. **Attach Policy to IAM Role:**
   - Create another JSON file with the policy details, for example, `role-policy.json`.
   - Modify the placeholders with your S3 bucket names.

Execute the command to attach the policy:
bash
Copy code

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

   -

Step 5: Upload OVA File to S3 Bucket

1. **Open AWS Management Console:**
    - Navigate to the S3 service.
    - Select the S3 bucket created for VM imports.
2. **Upload OVA File:**
    - Click on the `Upload` button.
    - Add the OVA file to the upload queue.
    - Start the upload process and wait for it to complete.

Step 6: Import VM to AWS

1. **Create JSON Container File:**
    - Use the JSON code provided on the AWS documentation for the import task.
    - Save the JSON code in a file, for example, `containers.json`.
2. **Run Import Command:**
    - Open the command prompt.

Execute the import command using the JSON container file:
bash
Copy code

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

    -

Step 7: Monitor Import Process

1. **Check Status:**
    - Use the command provided on the AWS documentation to monitor the status of the import task.

Run the command with the task ID obtained from the previous step:
bash
Copy code

```
aws ec2 describe-import-image-tasks --import-task-ids
import-ami-1234567890abcdef0
```

    -
    - Continue to monitor until the status changes to `completed`.

Step 8: Create an EC2 Instance from the Imported Image

1. **Create Instance:**
    - Open AWS Management Console.
    - Navigate to the EC2 service.

    - Launch a new instance using the imported image.
2. **Connect to Instance:**
    - Obtain the public IP address of the new instance.
    - Use RDP to connect to the instance.

Final Step: Verification

1. **Verify VM Functionality:**
    - Ensure the VM runs as expected on AWS.
    - Confirm connectivity and functionality of applications.