

Step 1: Setup VPC:

"Search for Services, features, blogs, docs, and more" > Type VPC > Click "VPC"
Click "Create VPC":

Auto-generate = "SRW VPC Testing"

IPv4 CIDR block = "10.0.0.0/16"

IPv6 CIDR block = "No IPv6 CIDR block"

Number of Availability Zones (AZs) = "2"

Number of Public subnets = "2"

Number of Private subnets = "2"

NAT gateways = "in 1 AZ"

VPC endpoints = "S3 Gateway"

Click "Create VPC"

VPC only

VPC and more

Name tag auto-generation [Info](#)
Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.

Auto-generate
SRW VPC TESTING

IPv4 CIDR block [Info](#)
Determine the starting IP and the size of your VPC using CIDR notation.

10.0.0.0/16 65,536 IPs

IPv6 CIDR block [Info](#)
 No IPv6 CIDR block
 Amazon-provided IPv6 CIDR block

Tenancy [Info](#)
Default

Number of Availability Zones (AZs) [Info](#)
Choose the number of AZs in which to provision subnets. We recommend at least two AZs for high availability.

1 2 3

▶ Customize AZs

Number of public subnets [Info](#)
The number of public subnets to add to your VPC. Use public subnets for web applications that need to be publicly accessible over the internet.

0 2

Number of private subnets [Info](#)
The number of private subnets to add to your VPC. Use private subnets to secure backend resources that don't need public access.

0 2 4

▶ Customize subnets CIDR blocks

NAT gateways (1) [Info](#)
Choose the number of Availability Zones (AZs) in which to create NAT gateways. Note that there is a charge for each NAT gateway.

None in 1 AZ 1 per AZ

VPC endpoints [Info](#)
Endpoints can help reduce NAT gateway charges and improve security by accessing S3 directly from the VPC. By default, full access policy is used. You can customize this policy at any time.

None S3 Gateway

DNS options [Info](#)
 Enable DNS hostnames
 Enable DNS resolution

Cancel Create VPC

Preview

VPC [Show details](#)
Your AWS virtual network
SRW VPC TESTING-vpc

Subnets (4)
Subnets within this VPC

- us-east-1a
 - SRW VPC TESTING-subnet-public1-us-east-1a
 - SRW VPC TESTING-subnet-private1-us-east-1a
- us-east-1b
 - SRW VPC TESTING-subnet-public2-us-east-1b
 - SRW VPC TESTING-subnet-private2-us-east-1b

Route tables (3)
Route network traffic to resources

- SRW VPC TESTING-rtb-public
- SRW VPC TESTING-rtb-private1-us-east-1a
- SRW VPC TESTING-rtb-private2-us-east-1b

Network connections (3)
Connections to other networks

- SRW VPC TESTING-igw
- SRW VPC TESTING-nat-public1-us-east-1a
- SRW VPC TESTING-vpce-s3

Step 2: Create an EC2 to Import the AMI from:

"Search for Services, features, blogs, docs, and more" > Type EC2 > Click "EC2" Click "Launch Instance"

The following are to be used for initializing your EC2 instance:

Application and OS Images (Amazon Machine Image): Amazon Linux
Instance type: t3.medium
Key pair name > Create "new key pair"
Key pair name = srw_pem
Click "create key pair" (make sure you save this to your desktop) Network settings > click "edit"
VPC = SRW VPC
Subnet > choose a public subnet
Check > "Allow HTTPs traffic from the internet" and "Allow HTTP traffic from the internet"
Configure storage = 250 GiB
IAM = ec2-user-dev
Click "Launch instance"

Step 3: Create Security Keys:

Go to IAM > "Manage Access Keys" > Create "new access key" (copy down the key id and secret access key)

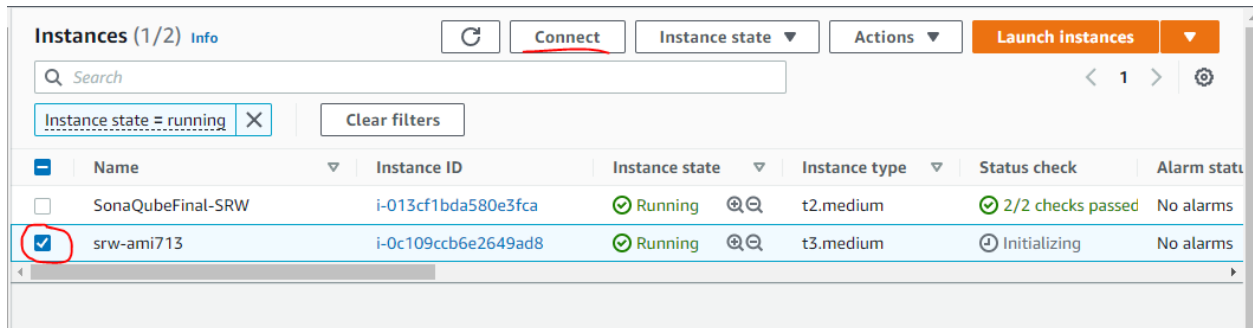
Download and save csv file in secure location as AWS will not show the secret key again.

Step 4: Log into EC2 and Create the AMI:

"Search for Services, features, blogs, docs, and more" > Type EC2 > Click "EC2"

Click "Instances (running)"

Check the instance state and click "connect"



Click "Connect"

> aws configure

Enter security key

Enter secret security key

Enter region name

Enter "json" for output format

>> touch trust-policy.json

>> vi trust-policy.json

Enter I on keyboard to enable edit mode

Insert json:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": { "Service": "vmie.amazonaws.com" },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals":{
```

```
        "sts:Externalid": "vmimport"
      }
    }
  }
]
}
```

Enter esc key when finished

Hold Shift key and press ZZ to exit json file

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

```
>> touch role-policy.json
```

```
>> vi role-policy.json
```

Enter I on keyboard to enable edit mode

Insert file:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::disk-image-file-bucket",
        "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": "*"
}
]
}

```

Click esc key when finished

Hold Shift key and press ZZ to exit json file

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

```
>> aws ec2 import-image --description "My SRW AMI" --disk-containers
Format=vmdk,Url=s3://epic-sandbox-srw/eports/export-ami-
0267d8751428919fd.vmdk
```