Aws approved /GoldenAMI

## **Risk level: Medium (should be achieved)**

Ensure that all the AWS EC2 instances necessary for your application stack are launched from your approved base Amazon Machine Images (AMIs), known as golden AMIs in order to enforce consistency and save time when scaling your application.

To determine if your EC2 instances are being launched using approved Amazon Machine Images (AMI), perform the following:

01.

aws ec2 describe-instances

--region us-east-1

--output table

--query 'Reservations[\*].Instances[\*].ImageId'

02.

The command output should return a table with the requested AMI IDs:

-------------------

|DescribeInstances|

+-----------------+

| ami-f5f41398 |

| ami-a8c439c5 |

| ami-b4cr745d |

+-----------------+

03.

Run **describe-images** command (OSX/Linux/UNIX) using the AMI ID returned at the previous as identifier to return the owner name for the AMI used to create the selected EC2 instance:

ws ec2 describe-images

--region us-east-1

--image-ids **ami-f5f41398**

--query 'Images[\*].ImageOwnerAlias'

04

The command output should return the owner name (e.g. self, amazon, aws-marketplace) for the selected AMI. In the following example output, the owner of the selected AMI is AWS:

[

"amazon"

]

If the value returned for the owner is different than **'self'**, the EC2 instance selected was deployed without using an approved/golden Amazon Machine Image (AMI), therefore the instance software configuration might not be stable and well-secured.

05

Repeat steps no. 3 and 4 to verify the AMI origin for the rest of the EC2 instances available in current AWS region.

06

Repeat steps no. 1 – 4 to verify the AMI origin for the EC2 instances launched in other AWS regions.

### **Remediation / Resolution**

To create golden/approved machine images and enforce your AWS administrators to launch EC2 instances using only these images, perform the following:

01

First, create the EC2 instance dependencies – the 2048-bit RSA key pair and the required security group:

1. Run **create-key-pair** command (OSX/Linux/UNIX) to set up a new RSA key pair in the selected AWS region:

aws ec2 create-key-pair

--region us-east-1

--key-name MyKeyPair

1. The command output should return the ASCII version of the private key and the key fingerprint. Save the content of your key, listed as the **KeyMaterial** parameter value, in a .pem file on your machine:

{

"KeyMaterial": "-BEGIN RSA PRIVATE KEY- ... -END RSA PRIVATE KEY-",

"KeyName": "MyKeyPair",

"KeyFingerprint": "ef:20:96:4a:5a:06:28 ... bb:20:0f:0f:c9:7b:g4"

}

1. Run **create-security-group** command (OSX/Linux/UNIX) to create a security group for the EC2 instance in the selected VPC. The following command example creates a security group called MySecurityGroup inside the VPC identified with the ID vpc-fb03eb9c, within the US East AWS region:

aws ec2 create-security-group

--region us-east-1

--group-name MySecurityGroup

--description "My EC2 Security Group"

--vpc-id vpc-fb03eb9c

1. The command output should return the new security group ID:

{

"GroupId": "sg-7550e90e"

}

02Run **run-instances** command (OSX/Linux/UNIX) to launch your base Linux/Windows EC2 instance. The following command example creates a new c4.large EC2 instance using an AMI with the ID ami-f5f41398 (Amazon Linux AMI 2016.03.1 base AMI), the RSA key pair and the security group created earlier, within the US East AWS region:

aws ec2 run-instances

--region us-east-1

--image-id ami-f5f41398

--count 1

--instance-type c4.large

--key-name MyKeyPair

--security-groups MySecurityGroup

03The command output should return the new EC2 instance configuration metadata:

{

"OwnerId": "123456789012",

"ReservationId": "r-05587b8359ad968fb",

"Groups": [],

"Instances": [

{

...

"EbsOptimized": false,

"LaunchTime": "2016-06-01T15:41:23.000Z",

"PrivateIpAddress": "172.31.12.90",

"ProductCodes": [],

"VpcId": "vpc-2fb56548",

"StateTransitionReason": "",

"InstanceId": "i-003b1c5834a73e5ae",

"ImageId": "ami-f5f41398",

...

"RootDeviceName": "/dev/xvda",

"VirtualizationType": "hvm",

"AmiLaunchIndex": 0

}

]

}

04Once the new EC2 instance is running, log in to the instance using the key pair created at step no. 1 and install the necessary software to run your application, secure the OS and the software stack and upload your application. Test the entire software stack to make sure that the EC2 instance qualifies for the golden image (AMI) then move to the next step.

05Run **create-image** command (OSX/Linux/UNIX) to create the approved/golden AMI using the EBS-backed instance created at step no. 2. Include the **–no-reboot** command parameter to guarantee the file system integrity for your new AMI:

aws ec2 create-image

--region us-east-1

--instance-id i-003b1c5834a73e5ae

--name "Approved/Golden Image"

--description "Web App Stack Production AMI ver. 1.4"

--no-reboot

06The command output should return the new Amazon Machine Image (AMI) ID:

{

"ImageId": "ami-e91ee384"

}

07Now that the golden AMI is ready for use, enforce your EC2 administrators to create instances from the new (approved) AMI only. To implement this restriction, perform the following:

1. Create a new policy document called **approved-ami-policy.json** and paste the following data (replace the highlighted details with your own details):

{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "Stmt1464769336975",

"Action": [

"ec2:RunInstances"

],

"Effect": "Allow",

"Resource": "\*",

"Condition": {

"ArnEquals": {

"ec2:ParentSnapshot":

"arn:aws:ec2:region::snapshot/snap-c5c0a423"

}

}

}

]

}

1. Run **create-policy** command (OSX/Linux/UNIX) to create the required IAM managed policy that will force your EC2 administrators to create instances using the golden AMI only:

aws iam create-policy

--policy-name ApprovedAMIPolicy

--policy-document file://approved-ami-policy.json

1. The command output should return the new IAM policy metadata (name, ID, ARN, etc):

{

"Policy": {

"PolicyName": "ApprovedAMIPolicy",

"CreateDate": "2016-06-01T16:12:43.987Z",

"AttachmentCount": 0,

"IsAttachable": true,

"PolicyId": "ANPAJIE7CHX7PBDD5COQQ",

"DefaultVersionId": "v1",

"Path": "/",

"Arn": **"arn:aws:iam::123456789012:policy/ApprovedAMIPolicy"**,

"UpdateDate": "2016-06-01T16:12:43.987Z"

}

}

1. Run **attach-group-policy** command (OSX/Linux/UNIX) using the IAM policy ARN returned at the previous step to attach the policy to the EC2 administrators group (if the command succeeds, no output is returned):

aws iam attach-group-policy

--policy-arn arn:aws:iam::123456789012:policy/ApprovedAMIPolicy

--group-name EC2Admins