AWS EC2 CLI commands

1. View Current Status of an Instance

The following “aws ec2 describe-instances” will display detailed information about all instances that are managed by you. The output will be in JSON format. aws ec2 describe-instances

If you have way too many instances, you can use the filter option to view a specific instance.

The following will display only the instance which has the “Name” tag set as “dev-server”.

# aws ec2 describe-instances --filter Name=tag:Name,Values=dev-server

..

..

"State": {

"Code": 80,

"Name": "stopped"

},

..

..

"InstanceId": "i-e5888e46",

..

From the above output, we can see that this instance is currently “stopped” and is not running.

2. Start an Instance

The following “aws ec2 start-instances” command will start the instance that is specified in the –instance-ids field.

This will also display the current state and the previous state of the instance in the output. As you see from the following output, previously this instance was “stopped” and now it is in “pending” state and will be started soon.

# aws ec2 start-instances --instance-ids i-dddddd70

{

"StartingInstances": [

{

"InstanceId": "i-dddddd70",

"CurrentState": {

"Code": 0,

"Name": "pending"

},

"PreviousState": {

"Code": 80,

"Name": "stopped"

}

}

]

}

If you want to start multiple instances using a single command, provide all the instance ids at the end as shown below.

aws ec2 start-instances --instance-ids i-5c8282ed i-44a44ac3

### 3. Stop an Instance

The following “aws ec2 stop-instances” command will stop the instance that is specified in the –instance-ids field.

As you see from the output, previously this particular instance was in “running” state and currently it is in “stopping” state and will be stopped very soon.

# aws ec2 stop-instances --instance-ids i-5c8282ed

{

"StoppingInstances": [

{

"InstanceId": "i-5c8282ed",

"CurrentState": {

"Code": 64,

"Name": "stopping"

},

"PreviousState": {

"Code": 16,

"Name": "running"

}

}

]

}

The following are the possible state name and state code for an instance:

* 0 is for pending
* 16 is for running
* 32 is for shutting-down
* 48 is for terminated
* 64 is for stopping
* 80 is for stopped

If you execute the above command on an instance that is already stopped, you’ll see both the previous state and the current state as stopped.

To stop multiple instances together, specify one or more instances ids as shown below.

aws ec2 stop-instances --instance-ids i-5c8282ed i-e5888e46

You can also force an instance to stop. This will not give the system an opportunity to flush the filesystem level cache. Use this only when you know exactly what you are doing.

aws ec2 stop-instances --force --instance-ids i-dddddd70

Be very careful when you are terminating an instance, as you can’t get your instance back once it is terminated. Terminate is not same as stop.

# aws ec2 terminate-instances --instance-ids i-44a44ac3

{

"TerminatingInstances": [

{

"InstanceId": "i-44a44ac3",

"CurrentState": {

"Code": 48,

"Name": "terminated"

},

"PreviousState": {

"Code": 80,

"Name": "stopped"

}

}

]

}

### 5. Add Name Tag to an Instance

The following “aws ec2 create-tags” command will add a new tag to the specified instance.

In this example, we are adding a tag with Key as “Department”, and it’s Value as “Finance”

aws ec2 create-tags --resources i-dddddd70 --tags Key=Department,Value=Finance

Now you’ll see that the new Tag has been added.

# aws ec2 describe-instances

..

"Tags": [

{

"Value": "Finance",

"Key": "Department"

},

{

"Value": "dev-server",

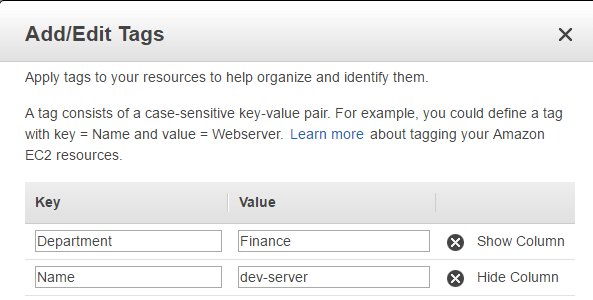
"Key": "Name"

}

],

..

You can also verify the TAG from the AWS Management Console GUI as shown below.



### 7. Launch a New EC2 Instance

The following command will create a new AWS EC2 instance for you.

This is equivalent to the “Launch Instance” that you’ll perform the AWS management console.

To launch an instance, use “aws ec2 run-instances” command as shown below.

# aws ec2 run-instances --image-id ami-22111148 --count 1 --instance-type t1.micro --key-name stage-key --security-groups my-aws-security-group

In the above command:

* –image-id Specify the image id for the AMI that you want to launch. You can browse the AWS marketplace and choose the correct image that is required for your project.
* –count Specify the number of instance that you want to launch from this image. In this case, we are creating only one new instance.
* –instance-type In this example, I’m launching this instance as a t1.micro type, which doesn’t use have CPU and RAM.
* –key-name Specify the name of the key pair that you want to use this with system. You should create your own key pair before launching your instance.
* –security-groups Specify the name of the security groups. You should create a security group with appropriate firewall rules that are required for your project.

The following is a sample full output of the above command, which display all the information about the newly launched instance.

{

"OwnerId": "353535354545",

"ReservationId": "r-d6668103",

"Groups": [

{

"GroupName": "my-aws-security-group",

"GroupId": "sg-6cbebe01"

}

],

"Instances": [

{

"Monitoring": {

"State": "disabled"

},

"PublicDnsName": "",

"KernelId": "aki-91afcaf8",

"State": {

"Code": 0,

"Name": "pending"

},

"EbsOptimized": false,

"LaunchTime": "2016-04-17T19:13:56.000Z",

"ProductCodes": [],

"StateTransitionReason": "",

"InstanceId": "i-44a44ac3",

"ImageId": "ami-22111148",

"PrivateDnsName": "",

"KeyName": "stage-key",

"SecurityGroups": [

{

"GroupName": "my-aws-security-group",

"GroupId": "sg-6cbebe01"

}

],

"ClientToken": "",

"InstanceType": "t1.micro",

"NetworkInterfaces": [],

"Placement": {

"Tenancy": "default",

"GroupName": "",

"AvailabilityZone": "us-east-1c"

},

"Hypervisor": "xen",

"BlockDeviceMappings": [],

"Architecture": "x86\_64",

"StateReason": {

"Message": "pending",

"Code": "pending"

},

"RootDeviceName": "/dev/sda1",

"VirtualizationType": "paravirtual",

"RootDeviceType": "ebs",

"AmiLaunchIndex": 0

}

]

}

If you get the following error message, then the instance type you’ve selected is not supported for this AMI. Change the instance type and try again.

# aws ec2 run-instances --dry-run --image-id ami-08111162 --count 1 --instance-type t1.micro --key-name MyKeyPair

A client error (InvalidParameterCombination) occurred when calling the RunInstances operation: Non-Windows instances with a virtualization type of 'hvm' are currently not supported for this instance type.

The following are additional parameters that you can pass with the “aws ec2run-instances” command

* –subnet-id Use the appropriate subnet id to launch a EC2 VPC instance
* –block-device-mappings file://mymap.json In this JSON file you can specify the volumes that you want to attach to the instance that you want to launch
* –user-data file://myuserdata.txt In this text file you can specify the userdata that need to be executed when the EC2 instance is launched
* –iam-instance-profile Name=myprofile You can also specify your IAM profile that you want to use while launching the instance

### Reboot an Instance (and General Options)

To reboot an instance, use “aws ec2 reboot-instances” command as shown below.

aws ec2 reboot-instances --instance-ids i-dddddd70

The are few options that you can use pretty much with most of the AWS EC2 cli commands.

For example, you can use “–dry-run” option pretty much with all the AWS EC2 cli command. As the name suggests, it will not really execute the command. This will only perform a dry-run and display all possible error messages without really doing anything.

### Change Instance Type

Before changing: In this example, the following instance is of type t1.micro

# aws ec2 describe-instances

..

"InstanceId": "i-44a44ac3",

..

"InstanceType": "t1.micro",

You can change the above instance to a different instance type.

For that, first stop the instance. Without stopping you cannot change the instance type.

aws ec2 stop-instances --instance-ids i-44a44ac3

The following “aws ec2 modify-instance-attribute” is used to change the instance type. In this example, we are changing the instance type to “m1.small”

aws ec2 modify-instance-attribute --instance-id i-44a44ac3 --instance-type "{\"Value\": \"m1.small\"}"

After changing, the following is the instance type.

# aws ec2 describe-instances

..

"InstanceId": "i-44a44ac3",

..

"InstanceType": "m1.small",

If an instance type is not supported for your particular image, you’ll get the following error message. In this example, t2.nano is not supported for this particular image.

### AWS EC2 Key Pairs

The following “aws ec2 describe-key-pairs” command will display all keypairs that you’ve created so far in AWS.

# aws ec2 describe-key-pairs

{

"KeyPairs": [

{

"KeyName": "prod-key",

"KeyFingerprint": "61:7c:f1:13:53:b0:3a:01:dd:dd:6c:90"

},

{

"KeyName": "stage-key",

"KeyFingerprint": "41:6c:d1:23:a3:c0:2a:0a:dc:db:60:4c"

}

]

}

To create a new Keypair use the following “aws ec2 create-key-pair” command. In this example, I’m creating a key pair with name “dev-servers”. I’ll be using this key-pair for all my dev instances.