EC2 CLI Commands

EC2 CLI - Introduction

Here is a video about simple AWS CLI commands to work with EC2.

With this video, we aim to get you familiar with EC2 CLI commands. Please just watch the video and try to understand general concept. It is used PowerShell in the video, so ignore some command unique to PowerShell like "select-string" ("grep" in BASH)
Then we will examine the basic EC2 CLI commands by applying them in the next lessons.

Describe-Instances

To display detailed information about all instances that are managed by your AWS account in JSON format, you can use `describe-instances` command as below.

```
$ aws ec2 describe-instances
```

- The output of this command will return you all the information about all the instances in your account.
- If you did not create an instance before, you'll get an empty JSON output like this.

```
{
    "Reservations": []
}
```

- Let's describe instances in our account to get detailed information about them.

```
$ aws ec2 describe-instances
```

- As you can see, the output consists of really detailed information and some times it's a little bit hard to handle.
- So, we can use `--query` option to display only the information we need about the instance.
- Here is an example of using this option.

```
$ aws ec2 describe-instances \
>     --query "Reservations[*].Instances[*].{Instance:InstanceId,AZ:Placement
    .AvailabilityZone,IPv4:PublicIpAddress,CurrentState:State.Name}"

```

- The above command with the given query option would return only the Instance Id, AZ, Public IP Address and State information.

```
[
    [
        {
            "Instance": "i-0c3d2f5873cf1dc2f",
            "AZ": "us-east-1d",
            "IPv4": null,
            "CurrentState": "terminated"
        }
    ],
    [
        {
            "Instance": "i-0475d2cf0786dca00",
            "AZ": "us-east-1d",
            "IPv4": "35.173.198.172",
            "CurrentState": "running"
        }
    ],
    [
        {
            "Instance": "i-0f727d7e59f0b21d7",
            "AZ": "us-east-1d",
            "IPv4": null,
            "CurrentState": "terminated"
        }
    ],
    [
        {
            "Instance": "i-0f2c467b0c8d0830d",
            "AZ": "us-east-1b",
            "IPv4": "54.174.23.221",
            "CurrentState": "running"
        }
    ]
]

```

- The output can also be returned as table format by using `--output table` option.

```
$ aws ec2 describe-instances \
>     --query "Reservations[*].Instances[*].{Instance:InstanceId,AZ:Placement
    .AvailabilityZone,IPv4:PublicIpAddress,CurrentState:State.Name}" \
>     --output table
-------------------------------------------------------------------------
|                           DescribeInstances                           |
+------------+--------------+-----------------+-----------------------+
|     AZ     | CurrentState |      IPv4       |       Instance        |
+------------+--------------+-----------------+-----------------------+
|  us-east-1d|  running     |  35.173.198.172 |  i-0475d2cf0786dca00  |
|  us-east-1b|  running     |  54.174.23.221  |  i-0f2c467b0c8d0830d  |
+------------+--------------+-----------------+-----------------------+

```

⚠ Avoid ! :

- Because EC2 is not a global service like S3, it's important to set the right region as default in configuration settings .
- If your default region and the region of the launched instances are not the same, the output will be an empty JSON.

Start-Instances

The `start-instances` command is used to start the instance(s) as below. More than one instance can also be started by a single code.

```
$ aws ec2 start-instances --instance-ids xxxxxxxxxxxx

$ aws ec2 start-instances --instance-ids xxxxxxxxxxxx yyyyyyyyyyyy zzzzzzzzzzzz
```

- To find the instance-ids, you can use `describe-instances` command that we learned in the last lesson.

```
$ aws ec2 start-instances --instance-ids i-0475d2cf0786dca00
{
    "StartingInstances": [
        {
            "CurrentState": {
                "Code": 0,
                "Name": "pending"
            },
            "InstanceId": "i-0475d2cf0786dca00",
            "PreviousState": {
                "Code": 80,
                "Name": "stopped"
            }
        }
    ]
}

```

- The output of this command also displays the current state and the previous state of the instance.
- Because this instance was previously in the “stopped” state, now it is in “pending” state and then will be started soon.

Tips: Instance State Codes

- 0 - Pending
- 16 - Running
- 32 - Shutting-down
- 48 - Terminated
- 64 - Stopping
- 80 - Stopped

Stop-Instances

The `stop-instances` command is used to stop the instance(s) as below. More than one instance can also be stoped by a single code.

```
$ aws ec2 stop-instances --instance-ids xxxxxxxxxxxx

$ aws ec2 stop-instances --instance-ids xxxxxxxxxxxx yyyyyyyyyyyy zzzzzzzzzzzz

```

- Example:

```
$ aws ec2 stop-instances --instance-ids i-0475d2cf0786dca00

{
    "StoppingInstances": [
        {
            "CurrentState": {
                "Code": 64,
                "Name": "stopping"
            },
            "InstanceId": "i-0475d2cf0786dca00",
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}

```

Tips: Instance State Codes

- 0 - Pending
- 16 - Running
- 32 - Shutting-down
- 48 - Terminated
- 64 - Stopping
- 80 - Stopped

Terminate-Instances

The `terminate-instances` command is used to terminate the instance(s) as below. More than one instance can also be terminated by a single code.

```
$ aws ec2 terminate-instances --instance-ids xxxxxxxxxxxx

$ aws ec2 terminate-instances --instance-ids xxxxxxxxxxxx yyyyyyyyyyyy
    zzzzzzzzzzzz
```

⚠ **Avoid ! :**

- Because the terminate is not the same as stop, be very careful when terminating an instance.
- Once an instance is terminated, you can't get it back again.

- Example:

```
$ aws ec2 terminate-instances --instance-ids i-0475d2cf0786dca00
{
    "TerminatingInstances": [
        {
            "CurrentState": {
                "Code": 48,
                "Name": "terminated"
            },
            "InstanceId": "i-0475d2cf0786dca00",
            "PreviousState": {
                "Code": 80,
                "Name": "stopped"
            }
        }
    ]
}

```

💡 **Tips: Instance State Codes**

- 0 - Pending
- 16 - Running
- 32 - Shutting-down
- 48 - Terminated
- 64 - Stopping
- 80 - Stopped

Run-Instances

The `run-instances` command is used to create new AWS EC2 instance(s). You can use different options with this command to specify the properties of the instance.

- For example, by using `--image-id` option like below, you can specify the Amazon Machine Image (AMI) type.

```
$ aws ec2 run-instances --imageid ami-0a887e401f7654935
```

- **ami-0a887e401f7654935** is the id of **Amazon Linux 2 AMI (HVM).**
- Now, let's look at a different example with more options.

```

 $ aws ec2 run-instances \
   --image-id ami-173d747e \
   --count 1 \
   --instance-type t2.micro \
   --key-name FirstKeyPair \
   --security-groups FirstSecGroup \
   --subnet Public-1a

```

In this example:

- The `--count` option means the number of instances that you want to launch from the specified image. In this case, only one new instance is being created.
- The `--instance-type` option defines general-purpose T2 instances that provide a baseline level of CPU performance with the ability to burst above the baseline.
- The `--key-name` option is the key-pair name you would like to use with this instance. Before starting the instance you should create your own key pair.
- The `--security-group` option specifies the name of the security groups you would like to use with this instance like a firewall.
- The `--subnet` option is the way for putting an instance to a subnet under a VPC. We will cover this subject later in the VPC subject.

⚠ **Avoid ! :**

- The **run-instances** command is used for launching EC2 instance.
- The **create-instance** command is used in AWS OpsWorks, to create an instance in OpsWorks stack.
- Be careful while using these 2 commands, which are convenient to be confused with each other.

Modify-Instance-Attribute

The `modify-instance-attribute` command is used to modify an attribute of the instance(s).

```
$ aws ec2 modify-instance-attribute \
  --instance-id xxxxxxxxxxxxxx \
  --instance-type "{\"Value\": \"m1.small\"\}"
```

- In this example, this command will change the **instance-type** to **m1.small.**
- Here is another example about the feature modifying process.

```
$ aws ec2 modify-instance-attribute
  --instance-ids xxxxxxxxxxxxxxx
  --disable-api-termination

```

- This command enables termination protection for the instance.
- If you want to disable the termination protection later, `no-disable-api-termination` can be used to do that.

Key-Pairs

Key-pair is the key to connecting to an EC2 instance.

For example, the following command will list all the keypairs.

```
$ aws ec2 describe-key-pairs
```

- You can also use the below commands to create and delete key-pairs on CLI:

```
$ aws ec2 create-key-pair --key-name MyFirstKey
```

```
$ aws ec2 delete-key-pair --key-name MyFirstkey
```