[Instructions and Commands] Understanding EC2 Metadata

Let us understand details about metadata related to EC2 Instance.

* All the actions that are performed while creating the instance are considered metadata.
  + Instance Type (Memory, CPU, and Instance Store).
  + Security Group
  + EC2 Key Pair
  + Private and Public IP Addresses as well as DNS Aliases
  + VPC and Subnet
* We can get all the metadata in the form of JSON by running the describe-instances command.
* Let us generate the JSON and review it to understand the structure of the metadata.

1. aws ec2 describe-instances \
2. --profile itvadmin \
3. --region us-west-1 > instances.json

[Instructions and Commands] Querying on EC2 Metadata

Let us understand how we can get specific attributes from metadata while describing instances using aws ec2 command.

* We can use **--query** to project the metadata related to specific attributes.
* As metadata is represented as JSON, we have to specify the attribute using the fully qualified JSON Path.
* Get only instance ids of all the instances.

1. aws ec2 describe-instances \
2. --query 'Reservations[\*].Instances[\*].{Instance:InstanceId,Status:State.Name}' \
3. --output json \
4. --profile itvadmin \
5. --region us-west-1

[Instructions and Commands] Filtering on EC2 Metadata

Let us understand how we can filter the data while describing instances using aws ec2 command.

* Get all the ec2 instances which are of type t2.micro

1. aws ec2 describe-instances \
2. --filters Name=instance-type,Values=t2.micro \
3. --output json \
4. --profile itvadmin \
5. --region us-west-1

* Get only instance id, type and status of t2.micro instances.

1. aws ec2 describe-instances \
2. --filters Name=instance-type,Values=t2.micro \
3. --query 'Reservations[\*].Instances[\*].{Instance:InstanceId,InstanceType:InstanceType,Status:State.Name}' \
4. --output json \
5. --profile itvadmin \
6. --region us-west-1

* Get only instance id, type and status of t2.micro instances in stopped state.

1. aws ec2 describe-instances \
2. --filters Name=instance-type,Values=t2.micro Name=instance-state-name,Values=stopped \
3. --query 'Reservations[\*].Instances[\*].{Instance:InstanceId,InstanceType:InstanceType,Status:State.Name}' \
4. --output json \
5. --profile itvadmin \
6. --region us-west-1

[Instructions and Commands] Using Bootstrapping Scripts

Let us understand how to take care of installing additional libraries or softwares as EC2 instances are created and started for the first time.

* Let’s terminate the existing **ec2demo** instance and create a new one with bootstrap script.
* Launch EC2 instance using Ubuntu 18.04
* Install Apache Web Server
* Install Python3 Pip
* Install AWS CLI using pip
* We can use this [document](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/user-data.html) as reference to prepare the ec2 instance as it is launched.

1. #!/bin/bash
2. apt update -y
3. apt install apache2 -y
4. apt install python3-pip -y
5. python3 -m pip install awscli

* Create instance using aws cli

1. aws ec2 run-instances \
2. --image-id ami-013f17f36f8b1fefb \
3. --count 1 \
4. --instance-type t2.micro \
5. --key-name keyname \
6. --security-group-ids sg-ID \
7. --user-data file://ec2\_user\_data.sh

[Instructions and Commands] Create an AMI

Let us understand how we can create AMI for existing EC2 Instance’s Volume. In our case, the EC2 instance is **ec2demo** and we will name the image as **ec2demoimage**.

* Go to EC2 Dashboard and then to Instances.
* Select the instance for which you want to create AMI.
* Go to Storage and click on the Volume.
* Create Snapshot for the Volume selected.
* Go to Actions and click on Create Image
* Creating AMI image from CLI

1. aws ec2 create-image \
2. --instance-id i-ID \
3. --name webAppAMI \
4. --description "A sample AMI with pre installed apache web server"

[Instructions and Commands] Validate AMI - Lab

Once the image is created, follow these steps to validate.

* Go to Images and create an instance using it.
* We have to go through the standard steps.
  + Choosing Instance Type
  + Configure Instance Details
  + Add Storage
  + Add or Choose Security Group
  + Launch with a new or existing Key Pair.
* Once the instance is launched, we will name it as **ec2demo1**.
* Wait until the instance is started and then connect via SSH. Once you connect via SSH, make sure to validate that we have Apache 2 as well as AWS CLI already setup and also Apache 2 is started on port 80.
* We can also use command line to create the instance using our AMI

1. aws ec2 run-instances \
2. --image-id ami-ID \
3. --count 1 \
4. --instance-type t2.micro \
5. --key-name keyname \
6. --security-group-ids sg-ID