**#Task 2**

**## Fetch the metadata in JSON format**

For this, I have used the Terraform outputs in the config files of a particular module, and then have included into main terraform output file and then have used command `terraform output –json` to get the JSON output of the metadata

**### The metadata that can be used as “key” that I am capturing is:**

* public\_ip
* public\_instance\_id
* public\_instance\_AMI\_id
* public\_instance\_type
* public\_subnet\_id
* public\_vpc\_security\_group\_ids
* private\_ip
* private\_instance\_id
* private\_instance\_AMI\_id
* private\_instance\_type
* private\_subnet\_id
* private\_vpc\_security\_group\_ids

For this, I have created a script under ‘scripts’ directory.

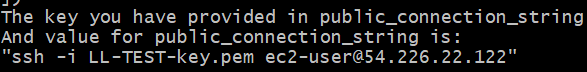
**Usage**:

bash instance\_metadata\_json\_processing.sh <Any key among above mentioned output keys>

**Sample input –**



**Sample Output –**



On broad level, this script essentially runs below commands

cd ..

terraform init

terraform plan

terraform apply -auto-approve

terraform output -json | jq ".$1.value"

Here, the $1 is the command line argument to the scipt.  
Hence, usage is –

bash instance\_metadata\_json\_processing.sh public\_connection\_string

TL;DR LOGS –

$ bash instance\_metadata\_json\_processing.sh public\_connection\_string

**Initializing modules...**

n\_string@WS-16641 MIa': bash instance\_metadata\_json\_processing.sh public\_connection\_string

**Initializing the backend...** instance\_metadata\_json\_processing.sh public\_connectio

**Initializing provider plugins...**

- Reusing previous version of hashicorp/local from the dependency lock file

- Reusing previous version of hashicorp/aws from the dependency lock file

- Reusing previous version of hashicorp/tls from the dependency lock file

- Using previously-installed hashicorp/local v2.1.0

- Using previously-installed hashicorp/aws v3.53.0

- Using previously-installed hashicorp/tls v3.1.0

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see

any changes that are required for your infrastructure. All Terraform commands

should now work.

If you ever set or change modules or backend configuration for Terraform,

rerun this command to reinitialize your working directory. If you forget, other

commands will detect it and remind you to do so if necessary.

**module.ssh-key.tls\_private\_key.key: Refreshing state... [id=173a990cc2331ee668a6ffc93c1b4347b649793f]**

**module.ssh-key.local\_file.private\_key: Refreshing state... [id=f59f0937590d0fe63d652e4779c441f9a5b20b5d]**

**module.ssh-key.aws\_key\_pair.key\_pair: Refreshing state... [id=LL-TEST-key]**

**module.networking.module.vpc.aws\_vpc.this[0]: Refreshing state... [id=vpc-05c64cb7c150d0b7b]**

**module.networking.module.vpc.aws\_eip.nat[0]: Refreshing state... [id=eipalloc-025492596c05951a3]**

**module.networking.aws\_security\_group.allow\_ssh\_priv: Refreshing state... [id=sg-0a2e2407085488394]**

**module.networking.aws\_security\_group.allow\_ssh\_pub: Refreshing state... [id=sg-07a9138bd68943e3c]**

**module.networking.module.vpc.aws\_internet\_gateway.this[0]: Refreshing state... [id=igw-0657b28807ca41b7a]**

**module.networking.module.vpc.aws\_subnet.public[0]: Refreshing state... [id=subnet-0771ecce8a1afeebd]**

**module.networking.module.vpc.aws\_route\_table.private[0]: Refreshing state... [id=rtb-012c62bea3bdf7734]**

**module.networking.module.vpc.aws\_route\_table.public[0]: Refreshing state... [id=rtb-0226926960f69c954]**

**module.networking.module.vpc.aws\_subnet.private[0]: Refreshing state... [id=subnet-07231c804628c5bf3]**

**module.networking.module.vpc.aws\_route\_table\_association.private[0]: Refreshing state... [id=rtbassoc-0d77d2efbdac5b742]**

**module.networking.module.vpc.aws\_route.public\_internet\_gateway[0]: Refreshing state... [id=r-rtb-0226926960f69c9541080289494]**

**module.networking.module.vpc.aws\_route\_table\_association.public[0]: Refreshing state... [id=rtbassoc-0dfd696378333b10f]**

**module.networking.module.vpc.aws\_nat\_gateway.this[0]: Refreshing state... [id=nat-05fd7b0697058367d]**

**module.networking.module.vpc.aws\_route.private\_nat\_gateway[0]: Refreshing state... [id=r-rtb-012c62bea3bdf77341080289494]**

**module.ec2.aws\_instance.ec2\_public: Refreshing state... [id=i-048637bc8db6d7170]**

**module.ec2.aws\_instance.ec2\_private: Refreshing state... [id=i-0d212c6b99767a428]**

No changes. **Your infrastructure matches the configuration.**

Terraform has compared your real infrastructure against your configuration

and found no differences, so no changes are needed.

**module.ssh-key.tls\_private\_key.key: Refreshing state... [id=173a990cc2331ee668a6ffc93c1b4347b649793f]**

**module.ssh-key.local\_file.private\_key: Refreshing state... [id=f59f0937590d0fe63d652e4779c441f9a5b20b5d]**

**module.networking.module.vpc.aws\_vpc.this[0]: Refreshing state... [id=vpc-05c64cb7c150d0b7b]**

**module.ssh-key.aws\_key\_pair.key\_pair: Refreshing state... [id=LL-TEST-key]**

**module.networking.module.vpc.aws\_eip.nat[0]: Refreshing state... [id=eipalloc-025492596c05951a3]**

**module.networking.module.vpc.aws\_subnet.public[0]: Refreshing state... [id=subnet-0771ecce8a1afeebd]**

**module.networking.aws\_security\_group.allow\_ssh\_priv: Refreshing state... [id=sg-0a2e2407085488394]**

**module.networking.aws\_security\_group.allow\_ssh\_pub: Refreshing state... [id=sg-07a9138bd68943e3c]**

**module.networking.module.vpc.aws\_subnet.private[0]: Refreshing state... [id=subnet-07231c804628c5bf3]**

**module.networking.module.vpc.aws\_route\_table.private[0]: Refreshing state... [id=rtb-012c62bea3bdf7734]**

**module.networking.module.vpc.aws\_internet\_gateway.this[0]: Refreshing state... [id=igw-0657b28807ca41b7a]**

**module.networking.module.vpc.aws\_route\_table.public[0]: Refreshing state... [id=rtb-0226926960f69c954]**

**module.networking.module.vpc.aws\_route\_table\_association.private[0]: Refreshing state... [id=rtbassoc-0d77d2efbdac5b742]**

**module.networking.module.vpc.aws\_route\_table\_association.public[0]: Refreshing state... [id=rtbassoc-0dfd696378333b10f]**

**module.networking.module.vpc.aws\_nat\_gateway.this[0]: Refreshing state... [id=nat-05fd7b0697058367d]**

**module.networking.module.vpc.aws\_route.public\_internet\_gateway[0]: Refreshing state... [id=r-rtb-0226926960f69c9541080289494]**

**module.networking.module.vpc.aws\_route.private\_nat\_gateway[0]: Refreshing state... [id=r-rtb-012c62bea3bdf77341080289494]**

**module.ec2.aws\_instance.ec2\_public: Refreshing state... [id=i-048637bc8db6d7170]**

**module.ec2.aws\_instance.ec2\_private: Refreshing state... [id=i-0d212c6b99767a428]**

No changes. **Your infrastructure matches the configuration.**

Terraform has compared your real infrastructure against your configuration

and found no differences, so no changes are needed.

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.

Outputs:

instance\_private\_ip = "172.20.20.23"

instance\_public\_ip = "54.226.22.122"

private\_connection\_string = "ssh -i LL-TEST-key.pem ec2-user@172.20.20.23"

private\_instance\_AMI\_id = "ami-0c2b8ca1dad447f8a"

private\_instance\_id = "i-0d212c6b99767a428"

private\_instance\_type = "t2.micro"

private\_subnet\_id = "subnet-07231c804628c5bf3"

private\_vpc\_security\_group\_ids = toset([

"sg-07a9138bd68943e3c",

])

public\_connection\_string = "ssh -i LL-TEST-key.pem ec2-user@54.226.22.122"

public\_instance\_AMI\_id = "ami-0c2b8ca1dad447f8a"

public\_instance\_id = "i-048637bc8db6d7170"

public\_instance\_type = "t2.micro"

public\_subnet\_id = "subnet-0771ecce8a1afeebd"

public\_vpc\_security\_group\_ids = toset([

"sg-07a9138bd68943e3c",

])

The key you have provided in public\_connection\_string

And value for public\_connection\_string is:

"ssh -i LL-TEST-key.pem ec2-user@54.226.22.122"