Practice 1: Global Infrastructure & Security

1) In the first exercise, I have done a script showing each AWS region with its availability zones.

   The commands used in the script are:

   #! /usr/bin/bash
   aws ec2 describe-availability-zones –availability-zones –region "us-west-2" –output table && aws ec2 describe-availability-zones –all-availability-zones –region "us-east-1" –output table

   First start the script with the shebang #!/usr/bin/bash, which will be read by the system and then run it.

   Then, I have written the following commands describing the specified instances.

   The syntax of AWS commands is: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
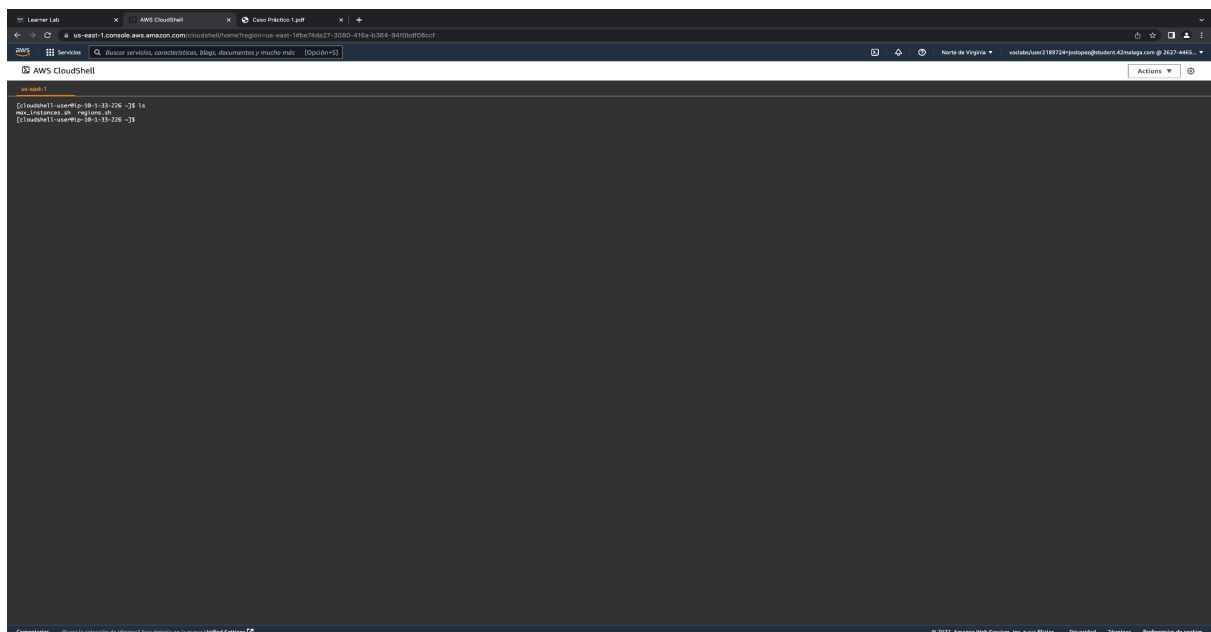
   This command would list all the instances from our default region defined for us.

   aws ec2 describe-instances

   To describe the instances in JSON from other regions, the region name can be specified on the command:

   aws ec2 describe-instances --region us-east-2

   With the command output table, I show the results in a table.

```
#!/usr/bin/env bash
aws ec2 describe-availability-zones --all-availability-zones --region "us-west-2" --output table && aws ec2 describe-availability-zones --all-availability-zones --region "us-east-1" --output table
```

"regions.sh" 2L, 217B   1,19   All


```
[cloudshell-user@ip-10-1-33-226 ~]$ bash regions.sh
```

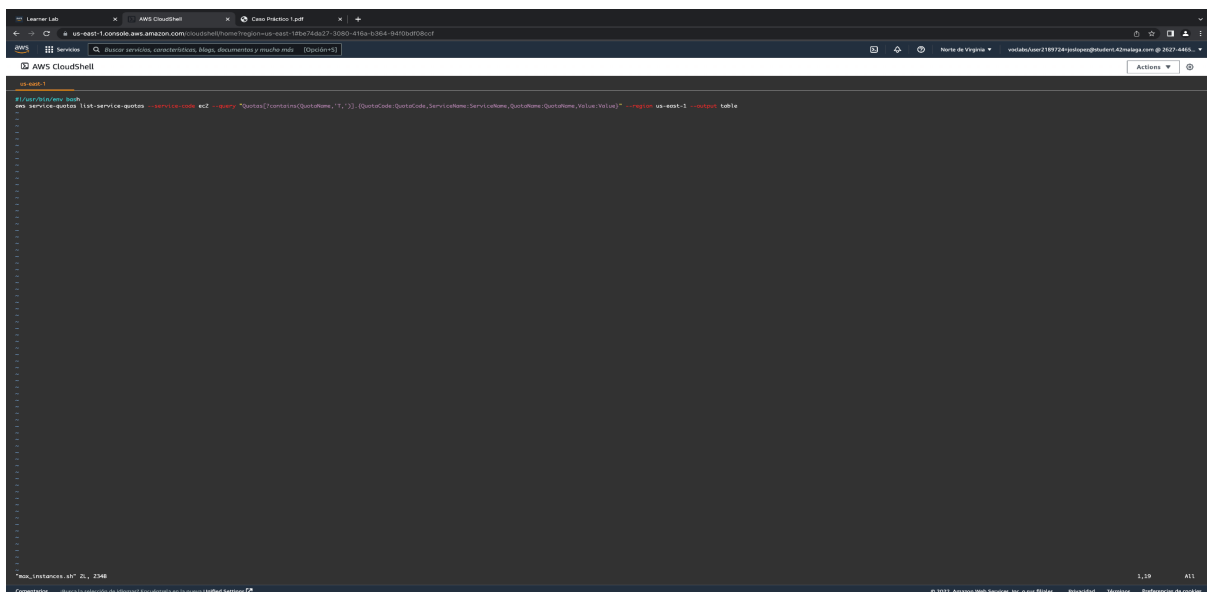| GroupName	NetworkBorderGroup	OptInStatus	ParentZoneId	ParentZoneName	RegionName	State	ZoneId	ZoneName	ZoneType
us-west-2	us-west-2	opt-in-not-required			us-west-2	available	usw2-az2	us-west-2a	availability-zone
us-west-2	us-west-2	opt-in-not-required			us-west-2	available	usw2-az1	us-west-2b	availability-zone
us-west-2	us-west-2	opt-in-not-required			us-west-2	available	usw2-az3	us-west-2c	availability-zone
us-west-2	us-west-2	opt-in-not-required			us-west-2	available	usw2-az4	us-west-2d	availability-zone
us-west-2-den-1	us-west-2-den-1	not-opted-in	usw2-az4	us-west-2d	us-west-2	available	usw2-den1-az1	us-west-2-den-1a	local-zone
us-west-2-lax-1	us-west-2-lax-1	not-opted-in	usw2-az3	us-west-2c	us-west-2	available	usw2-lax1-az1	us-west-2-lax-1a	local-zone
us-west-2-lax-1	us-west-2-lax-1	not-opted-in	usw2-az2	us-west-2a	us-west-2	available	usw2-lax1-az1	us-west-2-lax-1a	local-zone
us-west-2-lax-1	us-west-2-lax-1	not-opted-in	usw2-az4	us-west-2d	us-west-2	available	usw2-lax1-az2	us-west-2-lax-1b	local-zone
us-west-2-pdx-1	us-west-2-pdx-1	not-opted-in	usw2-az3	us-west-2c	us-west-2	available	usw2-pdx1-az1	us-west-2-pdx-1a	local-zone
us-west-2-phx-1	us-west-2-phx-1	not-opted-in	usw2-az2	us-west-2a	us-west-2	available	usw2-phx1-az1	us-west-2-phx-1a	local-zone
us-west-2-sea-1	us-west-2-sea-1	not-opted-in	usw2-az1	us-west-2b	us-west-2	available	usw2-sea1-az1	us-west-2-sea-1a	local-zone
GroupName	NetworkBorderGroup	OptInStatus	ParentZoneId	ParentZoneName	RegionName	State	ZoneId	ZoneName	ZoneType

2) In the second exercise, I have done a script showing the max number of EC2 type T instances I have permitted to create in the North Virginia region.
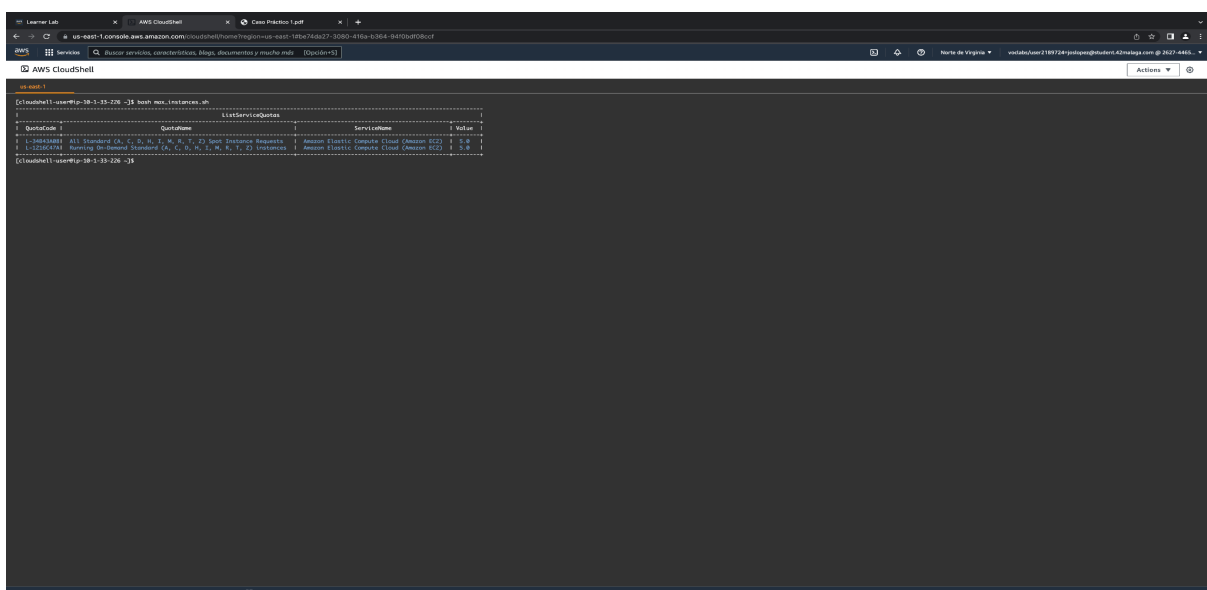
The commands used in the script are:

#!/usr/bin/env bash
aws service-quotas list-service-quotas --service-code ec2 --query
"Quotas[?contains(QuotaName,'T,')].{QuotaCode:QuotaCode,ServiceName:ServiceName,QuotaName:QuotaName,Value:Value}" --region us-east-1 --output table

With the command aws service-quotas, list-service-quotas, and service-code ec2 can be seen the list of service quotas there are in EC2. Command query enables one to do a faceted search and filter results based on document attributes.

Finally, I use the region command to define the region I want to see and the output table to show the results in a table.

Made by: José Antonio López Rosique, joslopez@student.42malaga.com