Question #1

Write a script in your language of choice to gather information about all of the instances in the current region. It should group them by the AMI’s that are in use, with information about the AMI's and how many EC2 Instances are using them. Format the output as a JSON object as below.

Assume that AWS Credentials are available in the environment.
If AMI’s are no longer available, null is an acceptable value for AMI specific items.
Do not assume that all AMI’s are owned by the current account.
Assume the environment being queried is a large account with lots of instances and AMIs.
Output results to stdout.

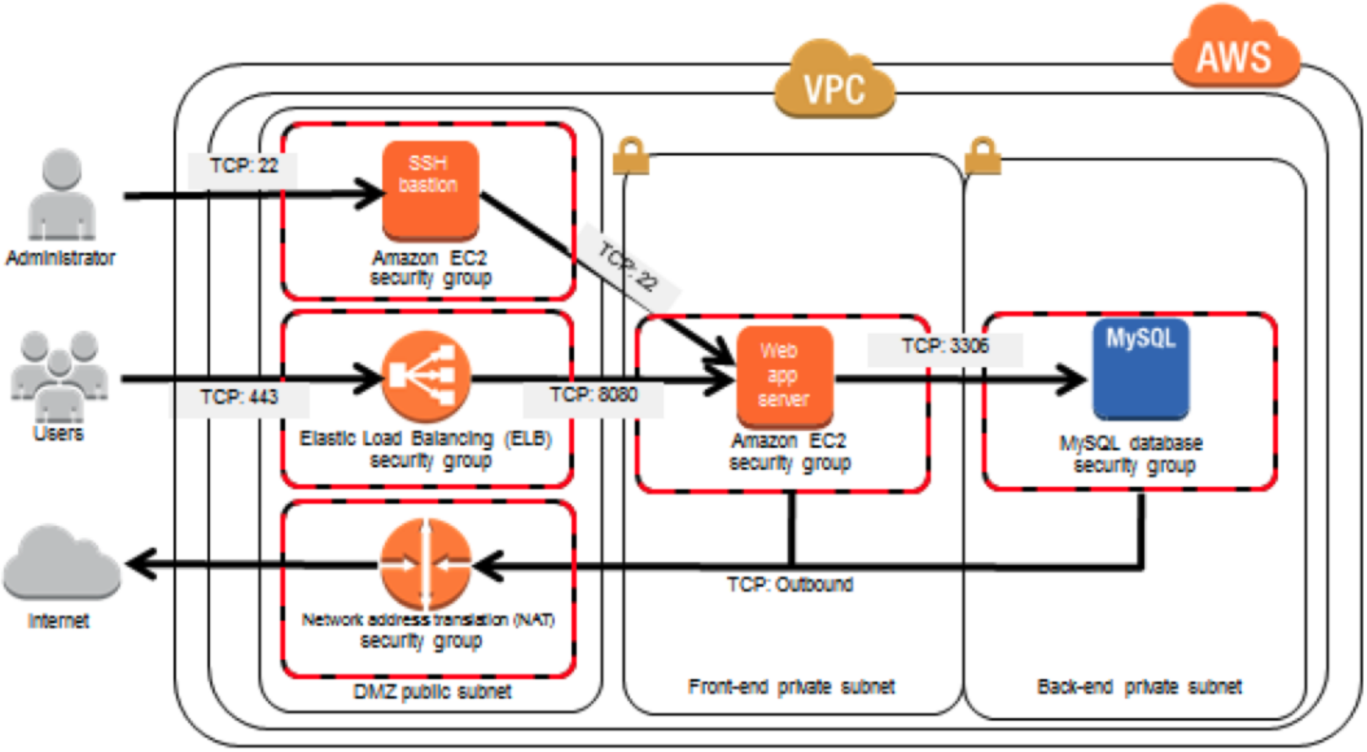
Example Structure

```
{
 "ami-0d5f069b7a75be450": {
                    "ImageDescription": "ubuntu-1804-encrypted-amd64-20190403",
          "ImageName": "ubuntu-1804-encrypted-amd64-20190403",
          "ImageLocation": "12345678901/ubuntu-1804-encrypted-amd64-20190403",
          "OwnerId": "12345678901",
          "InstanceIds": [
               "i-04f241953cfe0066d",
               "i-02439968b22cb6b8d"
          ]
 },
 "ami-c0464db9": {
          "ImageDescription": "Canonical, Ubuntu, 16.10, amd64 yakkety image buil
          "ImageName": "ubuntu/images/hvm-ssd/ubuntu-yakkety-16.10-amd64-server-2
          "ImageLocation": "099720109477/ubuntu/images/hvm-ssd/ubuntu-yakkety-16.
          "OwnerId": "099720109477",
          "InstanceIds": [
               "i-063e73a673eda7892"
```

```
        ]
    },
    "ami-1df0ac78": {
        "ImageDescription": null,
        "ImageName": null,
        "ImageLocation": null,
        "OwnerId": null,
        "InstanceIds": [
            "i-19be2ba7",
            "i-6a7a49dd"
        ]
    }
}
```

Question #2

Use your Infrastructure-as-Code tool of choice, e.g., Terraform, Cloudformation, etc. to provision the following set of resources in a given AWS account. Make sure you use best-practices, Nested Stacks in Cloudformation or Modules in Terraform. Note that in the given setup, incoming traffic on port 22 and port 443 goes to a Bastion host and an ELB, respectively to be forwarded to the web app server. The outgoing traffic from the web app server goes to a NAT to be sent to the Internet. Please create a separate security group for every component.


AWS
VPC
Administrator
Users
Internet
TCP: 22
SSH bastion
Amazon EC2 security group
TCP: 22
TCP: 443
Elastic Load Balancing (ELB) security group
TCP: 8080
Web app server
Amazon EC2 security group
TCP: 3306
MySQL
MySQL database security group
Network address translation (NAT) security group
TCP: Outbound
DMZ public subnet
Front-end private subnet
Back-end private subnet