Challenge 1

Public Layer: This layer consists of 3 public subnets. One on each Availability Zone.

Application Layer: This layer consists of 3 private subnets. one on each Availability Zone.

Database Layer: This layer consists of 3 private subnets. one on each Availability Zone.

**What goes in each of the layers?**

**Public Layer**  
The public (top) layer will host an internet-facing Elastic Load Balancer (ELB) and a Bastion host.  
The ELB is the entry point for your application and it directs traffic to your application servers.  
Note that the ELB needs to be available in all 3 availability zones. This will allow for high availability and redundancy, in case something happens to an entire availability zone. Our diagram only shows one instance of the ELB and you will only see one instance in the AWS console. However, behind the scenes AWS provisions multiple instances of the ELB based on what availability zones have EC2 instances behind that load balancer.  
The Bastion host (also known as Jump host) is the server that will allow you to connect to your application servers (or any other servers in the private subnets) via SSH.

**Application Layer**  
The second layer is the application layer, this is where your application servers live. In our case we have wrapped our application server with an Autoscaling Group. This will allow our application to scale up if more servers are needed or to recover in the case one of the availability zones is out of service. In the case where an entire availability zone is out of service the load balancer is smart enough to know that and it will scale up in a different availability zone.

**Database Layer**  
The third and last layer is the database layer. This is where the databases live. The only way to access these databases is by connecting to them from the application layer.

Challenge 2

|  |
| --- |
|  |
|  | /// The network interfaces on the instance. |
|  | /// </summary> |
|  | public static IEnumerable<NetworkInterface> NetworkInterfaces |
|  | { |
|  | get |
|  | { |
|  | var macs = GetItems("/network/interfaces/macs/"); |
|  | if (macs == null) |
|  | return null; |
|  |  |
|  | var interfaces = new List<NetworkInterface>(); |
|  | foreach (var mac in macs) |
|  | { |
|  | interfaces.Add(new NetworkInterface(mac.Trim('/'))); |
|  | } |
|  | return interfaces; |
|  | } |
|  | } |

Challenge 3

var obj, traverse;

obj = {

a1: {

b1: c1,

b1: d1

},

x1: {

y1: z1,

y1: a1

}

};

traverse = function(node, path) {

var pairs;

if (!(pairs = \_(node).pairs()).length) {

return [

{

keys: path,

value: node

}

];

} else {

};