SOLUTION

# Part 1: Q & A

1a Website down: stuff to check:

1. *Check the DNS servers (write IP address rather than the domain name on the address bar of the browser). If the IP address still doesn’t bring the site up there might be something wrong with the IP so we go to step 2. On the other hand, DNS/Domain name issues could be: the domain name has expired or very commonly as well that the SSL has expired.*
2. *IP issues. First, I would check if the website is down for everyone or just for me. If the site is down just for me it might imply that my ISP is blocking the IP address of the website server. But if the site is effectively blocked for everyone is more likely to be a firewall issue. So, I would check the firewall configuration. Are my firewall rules blocking inbound or outbound traffic? If the rules seem ok, I’d go to step 3.*
3. *Check load balancer. Is the load balancer up? If the load balancer is up we still have to check the load balancer configurations. Modern load balancers offer extra function (apart from distributing traffic to the application servers) such as security. Security against denial-of-service attacks. So, in case if the website being down, I would check the load balancer configurations. If they seem ok, I would go to point 4.*
4. *Server failure. There might be an issue with either the server hardware of the server software. If there is no sign of server failure, I’d go to point 5.*
5. *Updates on the hosting server operating system or the computer browser might cause the application code to fail. And this might cause the website to be down. If there have been no updates go to step 6.*
6. *Issues with 3rd party code. In modern web application we rely on 3rd party code. If this 3rd party is updated and we don’t keep uptodate with this code our website might go down.*

1b Website is back but slow.

*If a website is slow there might be many reasons for it but the first thing I would do (in a professional environment) is check loading times using monitoring tools. This monitoring tools will give you an interface to check loading times for the different elements of your website. I have seen in the past examples of larger images affecting the performance of a site.*

*So, I would compare the loading times between now (slow) and my benchmark loading times to see if something like the upload of an image or a change in code (JavaScript can create issues with speed) has happened.*

*The application described in this exercise includes a load balancer. A load balancer acts like a traffic police directing traffic to the different servers set up in the application. There are different load balancer configurations but we need to be sure the load balancer is directing traffic correctly to the different servers. It could happen for instance that 60 per cent of the traffic is directed to only one server. So, users hitting that server will notice poor performance.*

*If the load balancer configurations seem ok I would check to see we have caching installed or if it is working effectively. Is our application caching frequently retrieved information by users? If caching is not well set up, we might see issues of performance on our website.*

*Another aspect to consider for website performance if database overhead. Is the databased “tuned”?*

2. If I run the command /bin/chmod -x /bin/chmod will this run?

*No, I don’t think so. But I have a lot to learn on Linux command line so I might be wrong.*

*At least I have tried it and it doesn’t work for me. It gives me an error.*

*The syntax of the chmod command is:*

*chmod [options] <permissions> <filename>*

*So, I would have expected something like this if we wanted to remove execute permissions for the file file.txt:*

*chmod -x file.txt*

*Or if we have wanted to change the permissions of a user called “chmod” in a directory called “chmod” we would have to use a command like this:*

sudo chmod -x /bin/chmod

3. How long after an AWS security group is applied does it take for the rule to apply to all instances in that group?

No time at all, automatic.

Part 2 Scenarios

1 In custom made AWS VPC, what would be some reasons for lack of network connectivity.

1. *We forgot to attach the internet gateway to the VPC. No internet gateway no internet.*
2. *We didn’t edit our public route table so that our public instances are connected to the internet gateway. While we edit it, we need to be sure that we allow all Ips (destination 0.0.0/0) access to the internet gateway.*
3. *Lack of NAT Gateway or NAT instance in the private subnet. If we don’t include a NAT Gateway or NAT instance the applications hosted**in the private subnet will not have access to the internet.*
4. *NACL inbound rules impedes traffic to any of our EC2 instances.*
5. *Security rules inbound rules in our EC2 instances impede traffic.*
6. *NACL outbound rules impede traffic out of our EC2 instances.*

2. 3 EC2 instances sit behind an ELB. All devices are on a security group that has one rule only, which allows all traffic from the security group itself. However, hitting the ELB times out. Why?

*When creating the ELB we need to attach a security group opening up port 80 (TCP) in the inbound rules. I think that should be the solution. At least I have replicated the error by removing access to port 80 on the ELB security group and when hitting the ELB I was timing out.*

3. What’s best practice to allow your EC2 instances to talk to, say S3 or RDS.

*There are several best practices but one of them is to use an IAM role with S3 or RDS access credentials. The IAM role must be attached to the instance.*

*In the case of S3 another best practice is to implement the least privilege access so if our EC2 instance needs only to read from S3 we can use the AmazonS3ReadOnlyAccess policy.*

4. What steps would you follow to resolve the following problems?

Scenario 1 – Can’t connect to server

The customer phones up in a panic. One of their devs was fiddling with some AWS console settings and suddenly they have lost connectivity to their server.

Steps to try to resolve this problem:

1. I would log into the AWS console and check the status of the instance. If the instance has been terminated by mistake there is not much we can do. And any data stored on the instance will be lost. In addition, any attached EBS volume will be lost.
2. If the instance has been stopped accidentally, we can start it from the console and the user should be to recover connexion.
3. If the steps above do not solve the problem, I would then check the security groups and make sure that the inbound rules allow SSH or RDP (for Windows).
4. The next step would be to check the route table for the subnet. We need a route that connect to the internet gateway.
5. Then I would check the network access control list for the subnet. The network ACLs must allow inbound and outbound traffic.

Scenario 2 – Disk space is dangerously low!

**A few weeks later, the customer calls back and says that while just being logged in on the server for some routine maintenance, they accidentally noticed that the EBS volume, which houses all the images, is running at 92% disk utilization. They urgently need to expand their volume.**

*We can always modify the size of an EBS volume (elastic volume) in the console:*

1. *First go to your volume.*
2. *Actions (modify Volume).*
3. *Chose option to modify disk size or volume type.*
4. *Click Modify.*

*Size changes usually take a few seconds and take effect after a volume is in the Optimizing state.*

Scenario 2a – Why didn’t we know about disk space?

Following the disk space scare, they want to know why they weren’t aware of this happening and how they can be notified of this in the future.

*We can set up a CloudWatch alarm to warn us about disk space. The disk space alarm is not set up as a default metric on the console so you have to install on the instance a CloudWatch agent. We will have to configure that instance to push metrics like free disk space to CloudWatch.*

*Then we will be able to set up an alarm on that particular metric in CloudWatch. We will need to locate that EBS volume on the console. The CloudWatch metrics can be seen there.*

Scenario 3 – We need a failover error page

All these issues have got them thinking: if our site goes down, we have no way of telling our users that something is wrong and that we’re working on it. How can the customer have a basic failover in place?

*This is wide question; but one of the possible solutions is to failover to a S3 bucket. We will need to have an active Route 53 hosted zone; our application should be on an EC2 instance and Route 53 will need to route website traffic to the EC2 instance.*

*On the S3 bucket we will need to build a static web page, that static web page will be our error page. If our infrastructure fails Route 53 will failover to that static (error) web page.*