Answer to question 2:

I understand this is a very broad question so I will answer this question in two parts: the first is what are my responses to the incident and then how I would attempt to fix the problem.

When a website or service is down my first response is always to manually verify if it's still down. If the service went up already I will check the logs and investigate what happened when the alert was triggered. If the site is still down, I will take a quick measurement. If I can fix the issue quickly, I will try to fix it as soon as possible. If I cannot identify the issue or if the issue may take a long time to fix, I will notify the project manager and the person responsible for this application before I move forward to start the fix. I will tell them the estimated scale of impact, possible causes, and where I will start my next step. I will keep the communication until the issue is resolved.

My plan to fix the issue depends on my observations. **Assuming no immediate deployment was attempted before the incident and the automated alerts ping to a general access point such as the home page of a website.** The most likely causes of such failure are either network issues or application issues.

Network issues are that the internet is not able to resolve our IP correctly or users cannot reach it. For example, DNS records may be polluted, SSL certificates may be expired, and AWS may be under an attack. Or it could just be simply that there are too many requests. Network issues are relatively easy to trace, but their solutions can be very different from cause to cause.

Application failures can also cause the home page of a website to go down because the backend is not able to respond to the front-end request in a proper or timely manner (i.e. 500 and timeout errors). If the backend is a single server then I will try to restart the server first. If the server doesn't start then I know there is a system problem. I will check the server start logs. Usually, this is caused by system failures such as the file system being full or the local clock being off. If the backend is a cluster of servers then the problem is usually from external dependencies. For example, the database connection pool is used up, or the authentication service stopped working.

Again, servers can go down for many reasons. Above are just very general steps based on very broad assumptions. I'm interested to discuss more if you have more specific questions.