Question 1 Pt.1 (Everything Minus the After-Action Report)

1. ***Describe the steps that you would take to begin debugging the problem, to determine what might be going on.***
2. Check any Application Monitoring Dashboards for anything that might lead us in the direction of the issue
   1. Make sure there are no failing health checks for the infrastructure
      1. Make sure all configured services are functioning as normal and no health checks are failing or in the RED.
3. Check that DNS resolution is working correctly
4. Instead of connecting how most end users are connecting. Attempt to login to the servers as company admin through the bastion host.
   1. Run commands to verify functionality of the server.
      1. If we are reciveving errors from any commands. Record them, and investigate further
   2. Check services and verify all are running up to standard
5. Confirm that there were no changes around the time of incident to the servers.
6. Check the status of the ALB.
   1. Per the scenario described it appears traffic is only being forwarded to wpserver1 & not wpserver2 so this would also correlate with the step I would attempt here which would be: Checking the configuration of the ALB “Target Group” to confirm that it is configured correctly.
7. NOTES FOR THIS SECTION: Definitely had to do a decent amount of research specifically regarding bastion hosts and also ALB configuration/best practices.
8. ***What metrics, monitoring, or other tooling/processes would you expect to have available within your monitoring system to identify and validate the infrastructure is performing in a functional, healthy state?***
   * + 1. Typical system monitoring tools
          1. CPU, Disk, Mem, Network monitoring
          2. Ping to determine whether a machine is actually responding to request
       2. Service monitoring
          1. Some way to monitor whether services are running or not depending on the setup.
       3. Application Monitoring & Backend Database monitoring
          1. Someway to monitor typical activity for the application

This can include user logins or some other typical activity that the application/server performs

* + - * 1. Monitor and verify functionality of the Database
      1. Something similar to a syslog or graylog server were multiple logs are stored in a centralized location for review in times like these.

1. ***What changes or improvements could be made to the Architecture to reduce the risk of a similar incident?***
   * + 1. The biggest thing that jumped out to me is the ALB target group which appears to lack redundancy? I dfeintely could be wrong here but it appears all traffic is only flowing to wpserver1 and not wpserver2. In this case that would mean if there is some sort of issue with wpserver1 , that traffic is not being shifted over to wpserver 2. This could be because wpserver2 is not included in the ALB target group. I would add wpserver 2 into the ALB target group so in the event of a failure in wpserver1, the ALB will start sending traffic to wpserver2.
       2. There also appears to just be one RDS 1 instance and no measures for redundancy are mentioned in the initial scenario. If there are no measures for redundancy currently, investigation for the most cost effective mult-az redundant solution should be investigated.

NOTES: PLENTY of research was done on all the above analysis. Not having actual industry experience required a good amount of research for specific situations. A lot of researching time was looking into best practices for three tier architectures as well as AWS best practices for specific services.

**Now that the issue was resolved, and the environment is back into a healthy state; complete the following:**

1. **Create an After-Action Report (Post-Mortem/Post-Incident Review). It should include, at minimum, the following sections (more is highly encouraged):**
   1. **High-level Summary**
   2. **Participants (include who would be invited to an After-Action Report Discussion, as well other stakeholders who were involved in the incident)**
   3. **Timeline**
   4. **Root Cause Analysis**
2. High Level Summary
   1. On April 6th at 4:35pm we received reports of users unable to access clientapp.com. Per our procedure a ticket was opened due to the reported outage. The ticket number for this outage is #3019974. Due to the failure of one server in combination with a failure in overall configuration , the application began to fail. The application remained in it’s failed state for approximately 35 minutes. Once the problem server was rebooted, the issue was then resolved.
      1. Resolution
         1. Upon analysis of the reported issue, several commands were ran to test the functionality of server wpserver1. Once it was determined that the server was not responding to commands the machine was then rebooted. Upon reboot of the server, the application appeared to yet again be functioning as intended.
3. Timeline

4/6/2022 – 4:35PM: Ticket #3019974 opened as a High Priority incident.

4/6/2022 – 4:40PM: Ticket is assigned to Tier 3 technician.

4/6/2022 – 4:43PM: Ticket is marked as in progress by Tier 3 technician.

4/6/2022 – 4:55PM: Tier 3 technician reboots wpserver1.

4/6/2022 – 5:05PM: Previously failed health checks appear to now be green.

4/6/2022 – 5:07PM: Tier 3 tech contacted customer and had customer confirmed that application is now accessible.

4/6/2022 – 5:10PM: Ticket #3019974 marked as resolved.

Root Cause Analysis

A failure of a singular server caused a disruption of service and inaccessibility of the application. If the ALB (Application Load Balancer) was properly configured in this instance it should be serving traffic to both wpserver1 & wpserver 2. Currently it appears traffic is only being served to wpserver1 therefore once that server failed, the traffic no longer was being served. In other words the purpose of having wpserver2 is not being utilized as it should be, especially in situations such as this.

Follow-Up Action(s)

1. Update the application load balancer target group to include both wpserver1 and wpserver2.

2. Confirm that wpserver2 is ready to handle user traffic. Apparently, it has not been in the load balancer pool and may need remediation before it is prepared for production traffic.

3. Apply Infrastructure as Code processes so that configurations can be standardized, reviewed, and version controlled.

4. There also appears to just be one RDS 1 instance and no measures for redundancy are mentioned in the initial scenario. If there are no measures for redundancy currently, investigation for the most cost effective mult-az redundant solution should be investigated.

AAR Participants (roles):

Customer Manager

Incident Manager

Technicians & Dispatcher involved with the ticket