|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | **Coalfire** | | | | | |
| **After Action Report # 230** | | | | | | | | | |
|  | | | | | | **Ticket # 331346** | |
|  | | | | | | | | | | | |
| Summary | | | | | | | | | | | |
| On April 1, 2022, at 4:35PM a ticket 331346 was opened to report an outage of the clientapp.com application. Due to a configuration error, a failure of one server caused the application to fail. Users were unable to access services through the front end. | | | | | | | | | | |
|  | | | | | | | | | | | |
| Resolution | | | | | | | | | | | |
| After discovering the server, wpserver1, was unresponsive to commands at the console, the technician rebooted the server from the AWS console. The server rebooted successfully and began serving web pages normally. This returned application functionality to normal with all monitoring tools reporting ‘OK’. All monitoring tools showed ‘green’ and traffic monitoring for the site showed new user logins. | | | | | | | | | | | |
| Timeline | | | | | | | | | | | |
| 4/1/2022 – 4:35PM: Ticket 331346 opened  4/1/2022 – 4:40PM: Ticket acknowledged by tier 1  4/1/2022 – 4:55PM: Ticket acknowledged by tier 2  4/1/2022 – 5:15PM: Tier 2 technician rebooted wpserver1  4/1/2022 – 5:20PM: All monitoring dashboards show green  4/1/2022 – 5:30PM: Customer contact confirmed the application was reachable and working as intended.  4/1/2022 – 5:45PM: Ticket 331345 closed | | | | | | | | | | | |
| Root Cause Analysis | | | | | | | | | | | |
| A failure of a single server should not cause a failure for the application. Although the diagram provided in the documentation shows the Application Load Balancer (ALB) distributing traffic between both wpserver1 and wpserver2, the assumptions portion lists the following information:  1 ALB that has listeners in subnets *Public Subnet 1 & 2*   * + listens on port 443/TCP     - forwards traffic to the instance in subnet *WP Subnet 1*   The ALB is configured to distribute end user traffic only to server wpserver1 in WP Subnet 1. In this configuration, a failure of the availability zone or the server would cause an application failure. Wpserver2’s resources will never be utilized.  Based on system logs, the version of WordPress, and PHP, it appears the server experienced a known bug fixed in the latest version of WordPress. | | | | | | | | | | | |
|  | | | | | | | | | | | |
| Follow-Up Action(s) | | | | | | | | | | | |
| 1. Update the application stack to the latest compatible versions of Apache, PHP, WordPress, and related WordPress plugins. 2. Update the application load balancer target group to include both wpserver1 and wpserver2. 3. 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. 4. Apply Infrastructure as Code processes so that configurations can be standardized, reviewed, and version controlled. 5. Though not impacted by this incident, it is not clear whether the database instance, RDS 1, is configured for multi-Availability Zone (AZ) redundancy. Our recommendation would be to investigate the costs and benefits of implementing that multi-AZ redundancy for the database layer. | | | | | | | | | | | |
|  | |  | | |  | | | | | |
|  | | Incident Ticket Number(s): | | | Ticket # 331346 | | | | | |
|  | | Affected Agency(s): | | | CustomerX | | | | | |
|  | | Affected Service(s): | | | Clientapp.com | | | | | |
|  | | Configuration Item(s): | | | wpserver1 | | | | | |
|  | | Incident Priority: | | | Priority 2 | | | | | |
|  | | Date Write-Up Submitted: | | | 4/1/2022 | | | | | |
|  | | AAR Participants (roles): | | | Customer Liaison  Incident Manager  Technicians  Product Manager | | | | | |
|  |  | | | | | | | | | | | |
|  |  | | | | | | | | | | | |
|  |  | | |  | | |  | | | | | |
|  |  | | |  | | |  | | | | | |
|  |  | | |  | | |  | | | | | |
|  |  | | |  | | |  | | | | | |