# 

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Rank this week** | **Rank last week** | **Weeks on List** | **Risk Description** | **Severity** | **Likelihood** | **Risk Mitigation Plan** |
| **1** | **-** | **1** | Fail to respond and resolve change request within expected timeframe. | **Critical** | **High** | 1. Set up ticketing system notification service 2. Brief team members on what to do when there’s a change request |
| **2** | **2** | **3** | Peak traffic flow may exceed server load capacity, resulting in longer response times/unsuccessful responses | **Critical** | **Medium** | 1. Load balancing using multiple app servers on a single ec2 instance by using ports. 2. Monitor traffic flow using Cloudtopus and AWS Cloudwatch, team will create more ports if there is a high consistent network packets incoming over a period of time.. |
| **3** | **3** | **3** | Production server crashes during business hours and stops working. | **Critical** | **Low** | 1. Schedule for testing and maintenance on every Sunday. 2. Keeping daily backup of the website’s files (ec2 instance’s snapshot). If the site crashes, having a recent backup will ensure the site content will remain current. |
| **-** | **1** | **2** | Fail to obtain the required service levels set by the business: Zero downtime for the website on the Launch week | **Critical** | **High** | 1. Load balancing using multiple app servers on a single ec2 instance by serving the application on multiple ports. |