Observing Cloud Resources

*SRE Assessment Template*

# Categorize Responsibilities

|  |  |
| --- | --- |
| **Prometheus and Grafana Screenshots** | |
| Provide a screenshot of the Prometheus node\_exporter service running on the EC2 instance. Use the following command to show that the system is running: sudo systemctl status node\_exporter | |
|  | |
| **Host Metric**  **(CPU, RAM, Disk, Network)** | **Dashboard** |
| *[CPU]* |  |
| *[RAM]* |  |
| *[Disk]* |  |
| *[Network]* |  |
| **Responsibilities** | |
| 1. The development team wants to release an emergency hotfix to production. Identify two roles of the SRE team who would be involved in this and why. | |
| Roles will be:  smaller more frequent releases 🡪 smaller changes are faster and easier to fix, identify rollback, should an incident arise  isolated changes 🡪 these changes can be quickly identified and corrected if an incident arises  feature flags 🡪 this is probably the best way since it can be controlled, and it's isolated to a subgroup of users.  So, *we need to define ways to keep this hotfix for application to production more stable and make the release for this hotfix stable at production without going down. This to make sure for high availability for production and prevent hotfix from making production down. this is all to reduce downtime for production and satisfy end-user.*  So, 99.99% allows for approximately 52 minutes of downtime per year, or 13 minutes a quarter. If the development team's release causes an outage for 5 minutes,  then they would have 8 minutes left in the error budget for the quarter and can still release software. If the release causes 13 or more minutes of downtime,  releases are halted for the rest of the quarter. This error budget resets to 13 minutes in the next quarter.  So, roles will be **Release Manager** because it’s responsibilities are**:**   * Change management * Code releases * Ensures code has all dependencies satisfied * Executes the release, and rollback procedures * This role would use a CI/CD tool along with the source control tool to ensure the code could be released or rolled back, as well as verify the correct changes.   And **Monitoring Engineer To create dashboards and monitor application performance after releasing hotfix.** | |
| 2. The development team is in the early stages of planning to build a new product. Identify two roles of the SRE team that should be invited to the meeting and why. | |
| *1) Creating best practices, such as resilience to various single points of failure, that a developer team can employ when building a new product.*  *2) Documenting the dos and don’ts of particular infrastructure systems so developers can choose their building blocks wisely.*  *3) Providing early engagement consulting to discuss specific architectures and design choices in detail, and to help validate assumptions with the help of targeted prototypes.*  *4) Joining the developer team and participating in development work.*  *5) Codesigning part of the service.*  *That’s because Fixing architectural mistakes becomes more difficult later in the development cycle. Early SRE engagement can help avoid costly redesigns that become necessary when systems interact with real-world users and need to scale in response to service growth.*  *So,* **Infrastructure Engineer and Release Manager would be invited to this meeting because they are responsible for some things between development and operation:**  **Infrastructure Engineer:**   * 50% development tasks, 50% operations tasks * Planning/executing system patches/updates   **Release Manager:**   * Change management * Code releases * Ensures code has all dependencies satisfied * Executes the release, and rollback procedures * This role would use a CI/CD tool along with the source control tool to ensure the code could be released or rolled back, as well as verify the correct changes. | |
| 3. The emergency hotfix from question 1 was applied and is causing major issues in production. Which SRE role would primarily be involved in mitigating these issues? | |
| *Release Manager* | |

# 

# Team Formation and Workflow Identification

|  |
| --- |
| **API Monitoring and Notifications** |
| Display the status of an API endpoint: Provide a screenshot of the Grafana dashboard that will show at which point the API is unhealthy (non-200 HTTP code), and when it becomes healthy again (200 HTTP code). |
| *It becomes unhealthy at 15:34:05*  *And comes healthy again at 15:57:05* |
| Create a notification channel: Provide a screenshot of the Grafana notification which shows the summary of the issue and when it occurred. |
|  |
| Configure alert rules: Provide a screenshot of the alert rules list in Grafana. |
|  |

# 

# Applying the Concepts

|  |
| --- |
| **Graph 1** |
|  |
| 4a. Given the above graph, where does it show that the API endpoint is down? Where on the graph does this show that the API is healthy again? |
| *API endpoint goes in pending state at 15:28:00*  *API endpoint goes down at 15:35:00*  *API endpoint goes healthy again at 15:37:00* |
| 4b. If there was no SRE team, how would this outage affect customers? |
| *It will make the application down and no clients can access the application specifically it’s backend because the end point isn’t healthy anymore and it doesn’t give 200 HTTP status code. so no one can access the app or insert any data on app because it becomes down and there’s no one can handle issue and troubleshoot the endpoint except SRE team or monitor the app and its performance.* |
| 4c. What could be put in place so that the SRE team could know of the outage before the customer does? |
| *Alert rules can notify SRE team on configured Slack channel or through mail that the app comes down and end point is not healthy anymore so it will affect end users and no one can access the app anymore.*  *So SRE team will know before end users that app is down and they will troubleshoot it and handle this outage as soon as possible to don’t attract end user attention when accessing the app and notice it’s down. they may troubleshoot it before end user notices anything. After the alert sent to them on Slack channel or mail.*  *That’s all by configuring this alerts rules on Grafana platform to send these alerts or mails.* |

|  |
| --- |
| **Graph 2** |
|  |
| 5a. Given the above graph, which instance had the increase in traffic, and approximately how many bytes did it receive (feel free to round)? |
| *Instance had the increase in traffic is “10.0.0.68”*  *The num of bytes it received is 4800 bytes* |
| 5b. Which team members on the SRE team would be interested in this graph and why? |
| *Monitoring Engineer*  *Because he is the most interested in the dashboarding/alerting features of other types of monitoring software.*  *And he also usually first to know of an incident on instance.* |

# 

# 