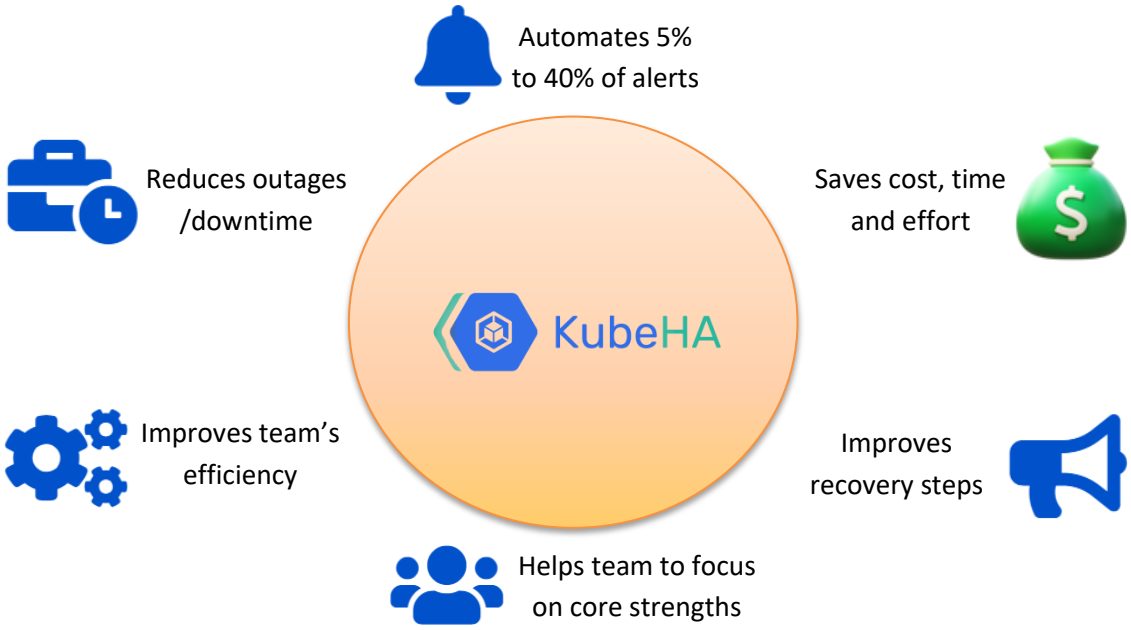
Observability and Monitoring

An alert recovery automation tool plays a crucial role in enhancing observability and monitoring practices by automating response actions when alerts are triggered.

KubeHA, an ultimate alert recovery automation tool enhances observability and monitoring by automating alert responses, reducing resolution times, and providing a more proactive approach to maintaining system reliability and performance.

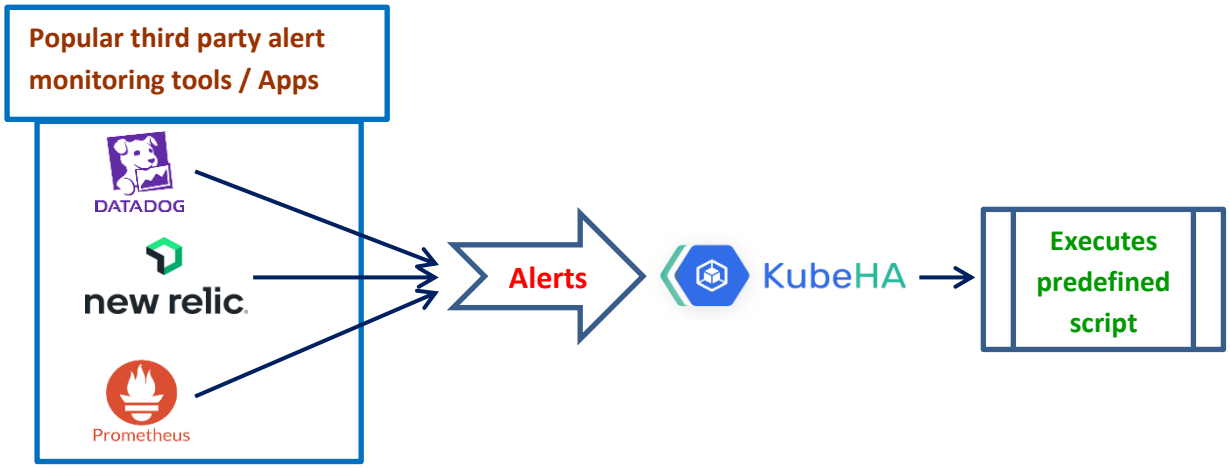
KubeHA provides flexibility to the users to automate recovery steps of alerts coming either from Kubernetes clusters or virtual machine clusters.

A. KubeHA's astonishing benefits:



B. KubeHA's unmatched features:

Below are some key aspects of how KubeHA can be utilized in the context of observability and monitoring:



1. **Integration with Monitoring Systems:**
   KubeHA gets integrated seamlessly with existing observability and monitoring systems. KubeHA has integration ready interfaces with popular third party alert monitoring tools: Datadog, New Relic and Prometheus. It can receive alerts from these tools. KubeHA can also receive alerts directly from any other application.
2. **Automated Alert Response:**
   KubeHA can be configured to respond automatically to specific types of alerts. For instance, it can execute predefined scripts to resolve common issues without manual intervention. The alerts can be automated by writing scripts in ShellScript/Python/Ruby.
3. **Reduced Mean Time to Resolution (MTTR):**
   By automating the response to alerts, KubeHA can significantly reduce the time it takes to address incidents. This is critical for maintaining system reliability and minimizing the impact of issues on end-users.
4. **Dynamic Response Playbooks:**
   Configure dynamic response playbooks that adapt to the nature of the alert. Different alerts may require specific actions, and KubeHA is flexible enough to accommodate various response scenarios.
5. **Alert Enrichment:**
   Enhance the alert recovery process by incorporating additional context or information related to the alert. This can include details about the affected components, recent changes, or historical data that aids in quicker problem resolution. KubeHA exports that information to the users and users can use them to perform additional actions/logic.
6. **Escalation and Notification:**
   KubeHA implements escalation policies within the tool to ensure that if an automated response doesn't resolve the issue, appropriate notifications are sent to the relevant personnel for manual intervention.
7. **Post-Incident Analysis:**
   KubeHA provides capabilities for analysing the actions taken during the incident recovery process. KubeHA stores the results and it is immutable. This information is valuable for post-incident reviews, audits, and continuous improvement of monitoring and response procedures.
8. **Adaptive Learning and Self-Optimization:**
   Over time, KubeHA learns from alerts and user interactions, improving its ability to handle similar situations in the future(upcoming AI). This adaptive learning enhances the efficiency and accuracy of automated responses.
9. **Security Incident Response:**
   KubeHA can be extended to security incident response scenarios. Automated actions helps contain and mitigate security threats, responding rapidly to potential breaches and vulnerabilities.
10. **Compliance and Auditing:**
    KubeHA complies with industry regulations and standards, hosted on AWS, uses DB encryption. Additionally, it offers auditing capabilities, allowing organizations to track changes, actions taken, and responses for compliance purposes.
11. **User-Friendly Configuration and Monitoring:**

KubeHA provides a user-friendly interface for configuring automation rules and monitoring the status of automated responses. This ensures that operations teams can easily manage and optimize the automation processes.

12. **Automation across clusters at a central place:**
    Automation of all the alerts coming from multiple third party monitoring tools(Datadog, New Relic, Prometheus, Apps) across all the clusters are at one place. It helps users to significantly improve team's efficiency.

C. KubeHA's Use Cases:

Below are some very basic KubeHA's use cases:

Alerts coming from Kubernetes Clusters:

1. **Disk got full**
   When alert "disk-gke-zn6 got full" is triggered from Gke cluster, it reaches to third party monitoring tool(say Datadog/New Relic/Prometheus). This alert gets forwarded to KubeHA's webhook. KubeHA finds the configured response actions(say delete older logs) written in the script for the alert and executes the response actions by login into Gke cluster.
2. **Node is not ready for more than 15 mins**
   When alert "node-gke-zn2 is not ready for more than 15 mins" is triggered from Gke cluster, it reaches to third party monitoring tool(say Datadog/New Relic/Prometheus). This alert gets forwarded to KubeHA's webhook. KubeHA finds the configured response actions(say isolate the node) written in the script for the alert and executes the response actions by login into Gke cluster.
3. **PVC size is getting filled up**
   When alert "Pvc-aws-zn2 is getting filled up" is triggered from Aws cluster, it reaches to third party monitoring tool(say Datadog/New Relic/Prometheus). This alert gets forwarded to KubeHA's webhook. KubeHA finds the configured response actions(say increase pvc size by 5Gb) written in the script for the alert and executes the response actions by login into Aws cluster.
4. **Pod is taking high CPU(>90%)**
   When alert "pod-7fb96c846b-lvgg5 is taking high CPU(>90%)" is triggered from Azure cluster, it reaches to third party monitoring tool(say Datadog/New Relic/Prometheus). This alert gets forwarded to KubeHA's webhook. KubeHA finds the configured response actions(say scale the pod or delete the pod, depending upon other parameters in alert) written in the script for the alert and executes the response actions by login into Azure cluster.

Alerts coming from virtual machine clusters:

5. **Disk usage is too high on a particular node:**
   When alert "high disk usage on ny-vm-node-2" is triggered from virtual machine cluster, it reaches to third party monitoring tool(say Datadog/New Relic/Prometheus). This alert gets forwarded to KubeHA's webhook. KubeHA finds the configured response actions(say delete older logs) written in the script for the alert and executes the response actions by login into virtual machine ny-vm-node-2.