Canary deployment: Past, present and potential future

DD2482 VT23, Essay

Jonas Sävås, jonassav@kth.se

I/We certify that generative AI, incl. ChatGPT, has not been used to write this essay. Using generative AI without permission is considered academic misconduct.

May 8, 2023

1 Introduction: DevOps and Deployment strategies

Development and Operations (DevOps) has immensely impacted the ways software is being deployed and published to consumers after the movement's debut in the early 2000s through a consensus of a dysfunctional industry.[11] The core of DevOps lies in the CI/CD pipeline which refers to Continuous Integration and Continuous deployment/delivery. The CI/CD pipeline aims to continuously integrate new features of the software and thereafter also continuously deploy them to the consumers. This naturally requires various different steps to take place where various steps are automated such as performing the initial unit tests where a solid infrastructure allows for automated deployments to the users if the new version passes the pipeline stages. The benefits of DevOps generally involve faster product shipments while also improving software stability and quality by automation of repetitive tasks where the team can put more focus on implementing new features.[8]

Introducing new features can be detrimental to a system that has undergone extensive changes and has not been tested under conditions that might only present themselves in production. DevOps avoids such big-bang releases by continuously deploying incremental changes but that is definitely not a one-stop solution for avoiding issues with deployed systems.[10] Another way that DevOps tries to maintain functionality is by adapting various deployment strategies to further minimize the impact of deploying new versions. One such deployment strategy to reduce the risk of major issues is Canary deployment which is a way to gradually rollout new releases.[7] Canary refers to an old practice of using Canary birds in coal mines in order to detect toxic gases where the deployment strategy adapts a similar approach, but not quite as gruesome. The main idea is to have a subset of your users test the new version where the traffic to the system gets split between the versions where some are presented with the new version while the rest is being served by the previous stable version. This percentage of traffic to the new version starts out quite low where its performance is monitored for any issues as the traffic is gradually increased if the new version seems ready. Once it has reached 100% the version is then fully released and if any critical errors occurred along the way the ability to instantly roll back to the previous version was available all along in order to minimize the impact on the user experience.

2 The Canary Evolution

Canary deployment has been used for many years within DevOps with notable mentions of companies such as Google, Facebook which recently changed to Meta and Netflix.[2] Sources for how early Canary deployments were carried out are very scarce but Danilo Sato discusses the earlier approaches in an article that Amazon references in their description of Canary deployment where a more hands-on approach took place during 2014. The deployment strategy then required the configuration for the servers to adopt the new version, directing a subset of the traffic and monitoring the deployment which was more or less a manual endeavor.[1] Manual processes carry the risk of being error-prone but also very time-consuming. It was not until the rise of cloud computing that specialized tools were developed and there is now a plethora of tools available that provide various benefits and drawbacks for different purposes. A system that is widely used today released back in 2014 for cluster orchestration is Kubernetes which can aid in automating various deployment strategies for containerized applications. Containers are essentially packaged software that includes the code and all its dependencies in order to achieve rapid setup on various machines that in turn increases its ability to quickly scale according to the demand.[12] Kubernetes then allows for your software to be distributed within your cluster of servers where it can distribute the traffic between different versions of the

application via its customizable load-balancer. The value of Kubernetes was quickly realized and it now has wide third-party support from cloud computing distributors such as Microsoft Azure and Amazon Web Services where Canary deployment can be integrated and load balancing automatically taken care of by Kubernetes.

A necessity for Canary deployments is the ability to monitor the deployed versions and quickly identify issues such as an increase in response time or error rate.[3] Kubernetes development over the years has also improved its monitoring capabilities where it can provide numerous metrics about your running cluster. However, the setup for quick identification providing graphs with automatic action has previously been tedious and complicated where multiple third party tools had to be connected. A popular tool that allows teams to quickly query data from their clusters is Prometheus which can be connected to graphical tools such as Grafana but this approach required many steps and a lot of manual configuration. Another problem of previous monitoring tools is that it could be very difficult or maybe even impossible to setup automatic notifications and automatic actions which creates a need for manual monitoring. Not being able to keep an eye on your systems at all times could therefore cause a large amount of downtime while also being error-prone, time-consuming, and of course costly. That is where todays more advanced monitoring tools come in.

Two tools that have risen to state-of-the-art status are DataDog which was initially released in 2018 and Dynatrace that released almost a year after in 2019. These tools have become popular for their ability to provide advanced monitoring capabilities while being customizable and in the case of Dynatrace impressively simple for the initial setup. Dynatrace makes use of their OneAgent software which sends the metrics to the Dynatrace servers for display in the Graphical User Interface (GUI) which can be customized to fit the team's requirements.[5] The installation process in the case of a Kubernetes cluster is narrowed down to a few commands that applies two YAML files and the agent is up and running. Taking this a step further is the ability to connect the metrics to Service Level Objectives (SLOs) in order for Dynatrace to be able to automatically notify the team if any issues have occurred or use their AI agent Davis in order to automatically notify the team if any issues have occurred or use their AI agent Davis in order to investigate for potential anomalies and notify the team before anything has even occurred.

Advanced monitoring solutions such as Dynatrace can aid in Canary deployment in various ways. First and foremost, the team needs to be able to determine the state of the different versions and use that information in the decisions to increase the percentage of traffic to the new version or completely rollback to the previous. Even the previous monitoring tools could provide this but what sets the recently developed tools like Dynatrace apart is the further automation of the deployment strategy. This comes in the form of making automatic decisions to based on the configuration and customized SLOs. Such SLOs could be connected to any of the gathered metrics such as a requirement for the new version to upkeep a certain success-rate with incoming requests or that the response time need to be within a certain interval. These decisions then range from automatically increasing the traffic to the new version if everything works as intended, automatically generating alerts for the team if or even before something happens, and in the case of errors that fails the SLOs an automatic remediation action can be issued to completely rollback to the previous version.[6] This does however require a lot of data from the system and sharing such data with complex analytical tools might not be suitable for all systems.

3 Challenges and Solutions

While Canary deployment can identify issues in production and quickly rollback in order to maintain functionality and only affect a subset of the users it does also pose various challenges in its execution. Besides the additional cost and potential for a cumbersome setup which has been mitigated by recent developments and support from cloud platforms, one of the greater challenges lies in deciding the target audience for testing the new version. This requires that valuable information can be gathered but also that the subjects are presented with a potentially broken version of your system in their critical workflow which could break their trust and simply cause them to switch to another platform. Another challenge of performing a successful Canary deployment is how long time should be spent in the testing phase and consequently how fast the traffic should be increased if the monitoring and observability tools do not report any critical errors.

Selecting your target audience for a Canary deployment could be done in various ways. Tomas Fernandez from Semaphore CI (2022) presents the advantages of Canary deployment and how one could go about choosing their target audience.[9] He mentions that this could be done completely randomly or deploy the Canary to a specific geographical region. This method does however still present the issue of ruining the trust of the users if an unstable version hinders their experience. In order to somewhat circumvent the issue, Fernandez mentions that you can then choose to use the strategy "follow-the-night" which refers to serving users with the new version during nighttime when the least users would be active. Another way to then include more users even earlier in the process would be to use an early adopter program where users can opt-in to test new features. However, the problem with these strategies are that they might not be able to capture the issues of the system in a full-scale environment with a low number of users. What big tech companies like Meta has done when releasing new features with Canary deployment is to apply a strategy called "Dogfooding". This strategy involves initially releasing the new features to internal users and employees which might solve the previous drawbacks for a company like Meta but might not be suitable for a smaller company.[9]

After selecting the testers for the Canary version the amount of time spent in each stage of the deployment before taking further release decisions is another important challenge that needs consideration. This naturally varies on a case-to-case basis but in the Google article "Canary analysis: Lessons learned and best practices from Google and Waze" (2019) they mention that in order for the monitoring data to be relevant "You need at least 50 pieces of time-series data per metric for the statistical analysis to be relevant.".[4] Following that baseline you would need to collect that information in each stage before either increasing the traffic or issuing an automatic remediation if any errors occurred.

4 Future Possibilities

Recent developments have introduced various tools to increase the support for deployment strategies such as Canary deployment. Recent advancements in AI has shown its immense potential for automation of various mundane tasks where Dynatrace is one of the tools that has integrated AI into how release decisions can be automated in Canary deployment. The future likely introduces more improvements and simpler ways of applying the DevOps pipeline which is discussed in the recent article "The Role of ML and AI in DevOps Transformation" by Gursimran Singh (2023).futureAI Singh mentions that automation will likely impact manual configuration where security aspects can be

increasingly analyzed to reduce the chances of issues and misconfiguration. This would likely come in the form of early detection where the AI systems can perform anomaly detection, deal with abrupt issues, and enhance the testing process in order to increase both functionality and security. With the recent release of OpenAIs ChatGPT there has been an intense discussion of the impacts of AI on software development as a whole where more mundane tasks can be automated and heavy integration into the development pipeline could take place where we might soon be able to see automatic code reviews and maybe even issue resolution where you define a problem with the source code and the AI provides a pull-request that potentially fixes the issue. However, only time will tell what is actually in store for DevOps in the future.

5 Conclusion

DevOps was born out of dissatisfaction with systems and principles for how software development should generally be handled. While providing many of the sought-after benefits the ability to maintain functionality when deploying new versions where testing could not fully simulate the production environment various deployment strategies were born. Canary deployment is one of the popular strategies that can be adapted for issues to surface while only impacting a subset of the users. The strategy does not come without its challenges where recent developments have both increased support and ease of setup while some challenges still remain. With the rise of AI and the potential improvements to software development such challenges that have previously required a production environment with a lot of users in order for the issues to surface can potentially be prevented by developments within advanced anomaly detection. The future has a lot in store for how deployments will take place with AI showing impressive potential to impact Canary deployment but also DevOps and software development as a whole.

References

[1] Danilo Sato. “CanaryRelease”. In: (2014). URL: https://martinfowler.com/bliki/CanaryRelease.html?ref=wellarchitected.

[2] Daniel Bryant. “A Comprehensive Guide to Canary Releases”. In: (2018). URL: https://blog.getambassador.io/cloud-native-patterns-canary-release-1cb8f82d371a.

[3] “Canary analysis: Lessons learned and best practices from Google and Waze”. In: (2019). URL: https://cloud.google.com/blog/products/devops-sre/canary-analysis-lessons-learned-and-best-practices-from-google-and-waze.

[4] “Canary analysis: Lessons learned and best practices from Google and Waze”. In: (2019). URL: https://cloud.google.com/blog/products/devops-sre/canary-analysis-lessons-learned-and-best-practices-from-google-and-waze.

[5] “Getting Started With The Basics of Dynatrace”. In: (2021). URL: https://www.linkedin.com/pulse/getting-started-basics-dynatrace-vishruth-harithsa.

[6] “How to automate Canary Release decisions with Dynatrace”. In: (2021). URL: https://www.dynatrace.com/news/blog/automate-canary-release-decisions-with-dynatrace/.

[7] Gopinath Rebala. “Five Advanced Deployment Strategies to Consider for Your DevOps Methodology”. In: (2021). URL: https://www.opsmx.com/blog/advanced-deployment-strategies-devops-methodology/.

[8] Fernando Almeida, Jorge Simões, and Sérgio Lopes. “Exploring the Benefits of Combining DevOps and Agile”. In: *Future Internet* 14.2 (Feb. 2022), p. 63. ISSN: 1999-5903. DOI: 10.3390/fi14020063. URL: http://dx.doi.org/10.3390/fi14020063.

[9] Tomas Fernandez. “What Is Canary Deployment”. In: (2022). URL: https://semaphoreci.com/blog/what-is-canary-deployment.

[10] Mitul Makadia. “9 Key Benefits of DevOps”. In: (2022). URL: https://www.business2community.com/business-intelligence/9-key-benefits-of-devops-02391855.

[11] Ian Buchanan. “History of DevOps”. In: (2023). URL: https://www.atlassian.com/devops/what-is-devops/history-of-devops#:~:text=The%5C%20DevOps%5C%20movement%5C%20started%5C%20to,of%5C%20dysfunction%5C%20in%5C%20the%5C%20industry..

[12] “What are Containers?” In: (2023). URL: https://cloud.google.com/learn/what-are-containers.