

	Bansilal Ramnath Agarwal Charitable Trust's Vishwakarma Institute of Information Technology	
	<b>Department of Artificial Intelligence and Data Science</b>	
<b>Student Name:</b> Agrawal Yash Pradeep		
<b>Class:</b> TY	<b>Division:</b> A	<b>Roll No:</b> 371005
<b>Semester:</b> 5th		<b>Academic Year:</b> 2022-23
<b>Subject Name &amp; Code:</b> Cloud Computing and Analytics		
<b>Title of Assignment:</b> Assignment-8 Study and explore – case study with devops and site reliability engineering.		
<b>Date of Performance:</b> 01/12/2022		<b>Date of Submission:</b> 02/12/2022

**Aim:** Study and explore – case study with devops and site reliability engineering.

**Problem Statement:** Study and explore – case study with devops and site reliability engineering.

### **Background Information:**

#### **Introduction:**

DevOps, which stands for Development and Operations is an approach through which developers work in collaboration with the operation teams. This sort of approach enables accountability for the end-to-end process considering all parts from requirement planning, development, testing, and production.

1. Increased Accountability :DevOps adoption concretely helps save time-to-market, a significant amount of money, faster result visibility, removed manual tasks by enabling automation and helps avoid delays and hindrances due to lack of communication. All this combined also consecutively leads to enhanced quality of produced software.
2. Prioritization of Work The key with DevOps is possession and accountability, and especially making sure that one team, comprising

developers, venture managers, enterprise analysts and testers, take up ownership for the entire process. Teams are now moving away from the conventional waterfall software program improvement cycles and adopting agile and lean methodologies. This allows features and requirements to be prioritised as appropriate, thereby lowering the time taken to finish tasks. It encourages the invention of issues in advance within the lifecycle, making them simpler, less expensive and much less time-taking to fix.

3. End-to-end Automation and Agility Introducing automation in multiple areas such as QA, Deployment, and Continuous Integration (IC) can greatly speed the development process and reduce human effort and the risk associated with it. Promoting the introduction of agile methods such as Kanban or Scrum can accelerate the cycle even further as they allow the team to monitor and prioritize the workflow, taking into account both planned and unplanned activities.
4. Higher quality software As the team is aware of the end-to-end process and is accountable for the same the collaboration and communication between different stakeholders is efficient and this leads to a better quality of software production.
5. Significant financial savings Studies show that adopting DevOps has helped companies save as much as 20 million US dollars per year which leads to more rapid software releases and also drastic reductions in business-impacting incidents due to smaller and faster releases.

### Case Study of Nordstrom company:

**About company:** Nordstrom, Inc. was established in 1901, more than a century ago. The company, headquartered in Seattle, Washington, was founded by John W. Nordstrom and Carl F. Wallin as a shoe shop. Clothing, accessories, purses, jewelry, cosmetics, and fragrances have all been added to the company's inventory since then.

**How Nordstrom adopted DevOps:** The operations team looked at shifting the paradigm from performance testing environments to adopting a development process with quality and testing embedded throughout the lifecycle. The team looked into adopting solutions and ideas which were being discussed in the ongoing DevOps movement and this was the onset of Nordstrom building Infrastructure as Code and Continuous delivery so testing happens on the go in

parallel with development. Along with this, the team decided to move on to an incremental development process, with frequent feedback and validation loops so that the team stays agile and is flexible to make changes quickly as well as to avoid the earlier mistakes like the “Personal Book” rewrite failure.

In 2013, Nordstorm took its first step towards a Cloud Native journey by launching a project focusing on Continuous Integration/Continuous Deployment. The team orchestrated a pipeline to enable CI/CD and chose Chef, an open source Configuration management tool to automate virtual IP creation, spinning of servers through code, and automatic load balancing. This helped achieve faster deployments and releases.

By 2015, Nordstorm integrated into a lot of its internal applications a monitoring tool called Splunk. It’s a tool that focuses on converting data into valuable insights that would help power security, IT and DevOps teams make better decisions. At Nordstorm, Splunk was initially used to monitor website performance – page loading times, uptime while keeping track of all the deployments and releases. They also used Splunk as a security layer. The entire Splunk deployment was managed like source code, with developers developing various UI dashboards and apps for Splunk’s internal usage. From automated builds to updates, Nordstrom used Chef to deploy, maintain, and configure Splunk systems all using a single chef cookbook, so when any environment updates were needed, devs pushed it to chef and would be automatically configured.

Having multiple environments for dev, test and production still acted as a hindrance and this pushed for the need to work in the cloud. Using Kubernetes to orchestrate Docker containers helped the company have all 1500 of their developers deploy code as applications running as containers in the Cloud. Now the applications could run anywhere from on-prem to the cloud which enabled portability and took minimal setup time. With the use of the cloud accessing resources was much faster and easier instead of having to acquire virtual machines on-premises with the added benefit of reduced risks. As teams began to run on Kubernetes clusters they immediately realized the benefits of 0 0 self-management of infrastructure and operations. With the growing cluster, the team was now able to build production-grade services quickly and easily without worrying about the underlying complication.

Adopting Kubernetes showed drastic reductions in deployment time, along with a 10x increase in CPU utilisation. Applications ran on around 40 VMs

which if run directly on the cloud without Kubernetes cluster management would scale to around 2500+ VMs which goes to show the reduction in operational costs.

The team adopted Prometheus for monitoring along with Grafana for the visualization of graphs and alerts. To go in hand with their monitoring solutions Nordstorm used FluentD for building a unified logging layer that pushed the logs to Elasticsearch which helped with efficient log aggregation.

### Outcomes:

With almost 2000 employees working in their IT sector, Nordstorm took a brave but necessary step towards optimizing cost and efficiency by adopting the DevOps culture. The adoption not only helped them make their engineering teams efficient but also helped them grow as a strong business entity. They also were able to adopt the devops culture into their business teams by encouraging constant feedback, quick and continuous improvement. Here are some of the perfect examples of how adopting DevOps helped Nordstorm survive the market:

- Reduced deployment cycles from 3 months to almost 30 minutes.
- More frequent releases with the upgrade from the previous 28 weeks to now 4 weeks.
- Higher quality updates with better tracking of issues at hand.
- Improved CPU Utilisation from 5x to 12x depending on workloads which meant increased Operations efficiency.
- Increase in employee happiness
- Building at scale, complex systems with larger configurations still would work with the same resources.
- Better overall planning and goal achievements

Conclusion: We studied devops and how Nordstorm company adopted devops.