

# DevOps Pipeline Presentation

---

- Gear5\_Grupparbete

YH Nackademin 2023



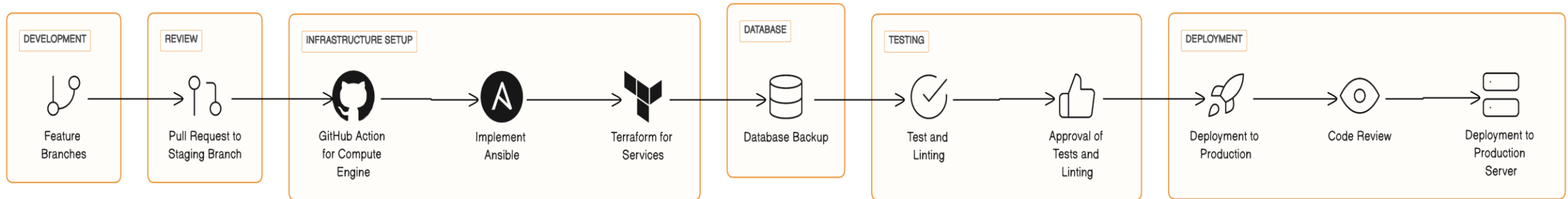
# DevOps Pipeline?

---

- En DevOps-pipeline är en automatiserad och kontinuerlig process som möjliggör en smidig och effektiv utvecklings- och leveranscykel för mjukvara. Dess betydelse i mjukvaruutveckling är att den underlättar samarbete mellan utvecklings- och driftsteam, automatiserar testning och distribution, och möjliggör snabba och pålitliga leveranser av mjukvaruförändringar till produktion.



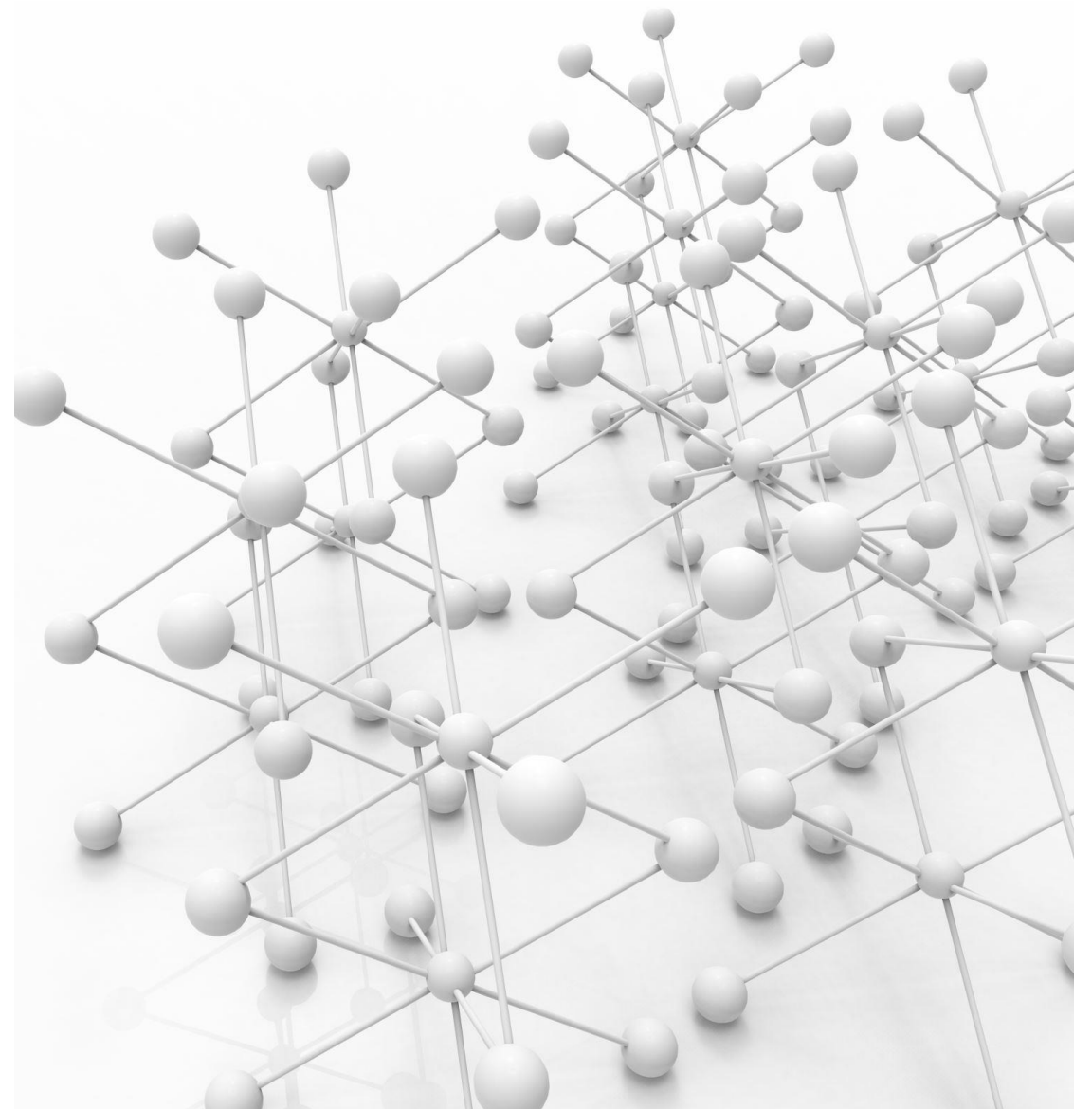
# Flödesschema av CI/CD-pipeline för produktionsserver.



# Feature Branches

---

- Utvecklingsarbetet sker i separata feature branches, vilket är en god praxis för att isolera ändringar och underlätta parallell utveckling.





# Pull Request till Staging Branch

---

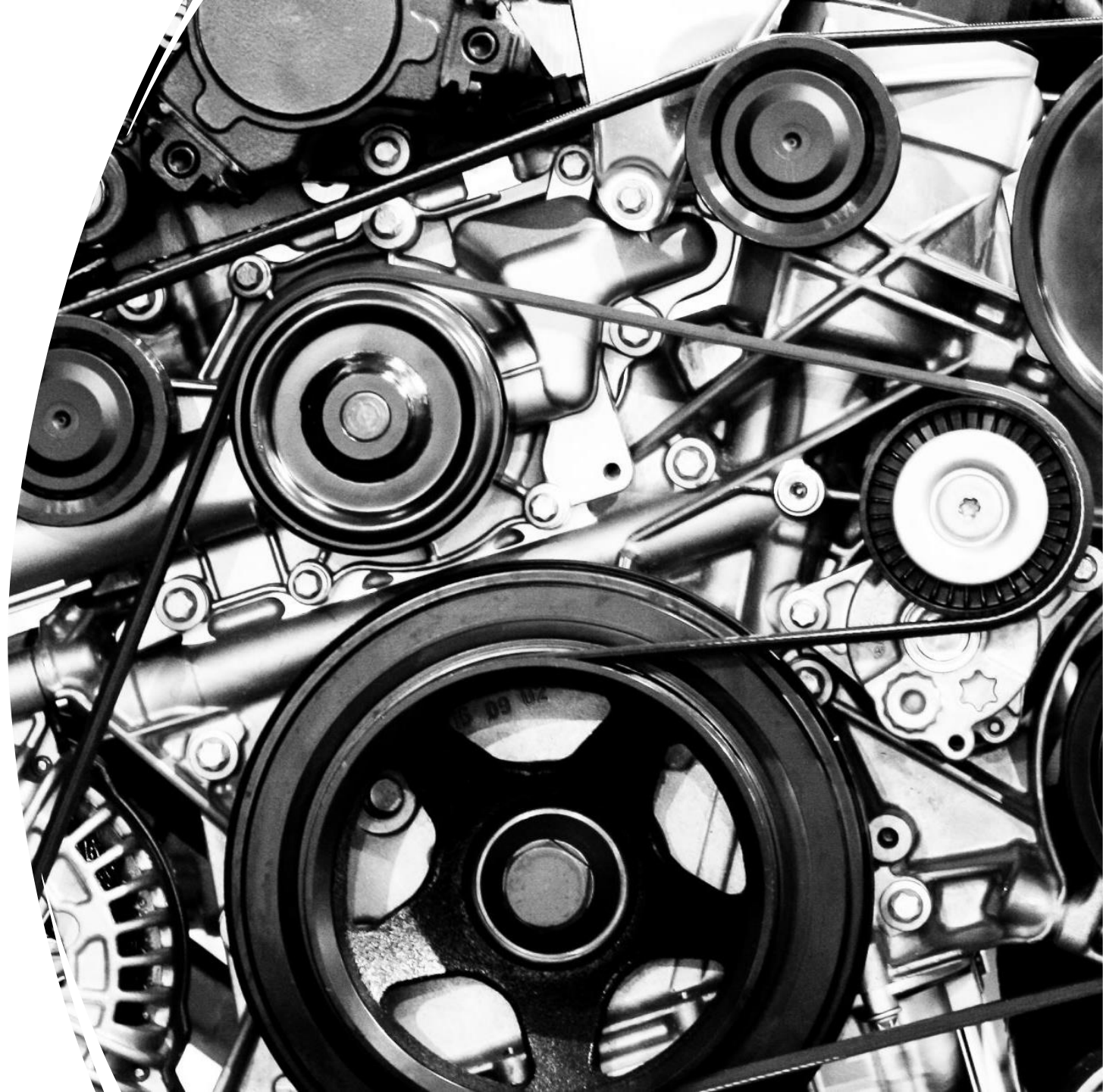
- När en funktion är klar, öppnas en Pull Request (PR) till staging-branchen. Detta ger en möjlighet till granskning och testning innan ändringarna integreras.



# GitHub Action för att skapa Compute Engine

---

- När PR mergas till staging, utlöses en GitHub Action. Denna åtgärd använder Terraform för att skapa en Compute Engine på Google Cloud Platform (GCP). Detta är ett exempel på Infrastructure as Code (IaC).

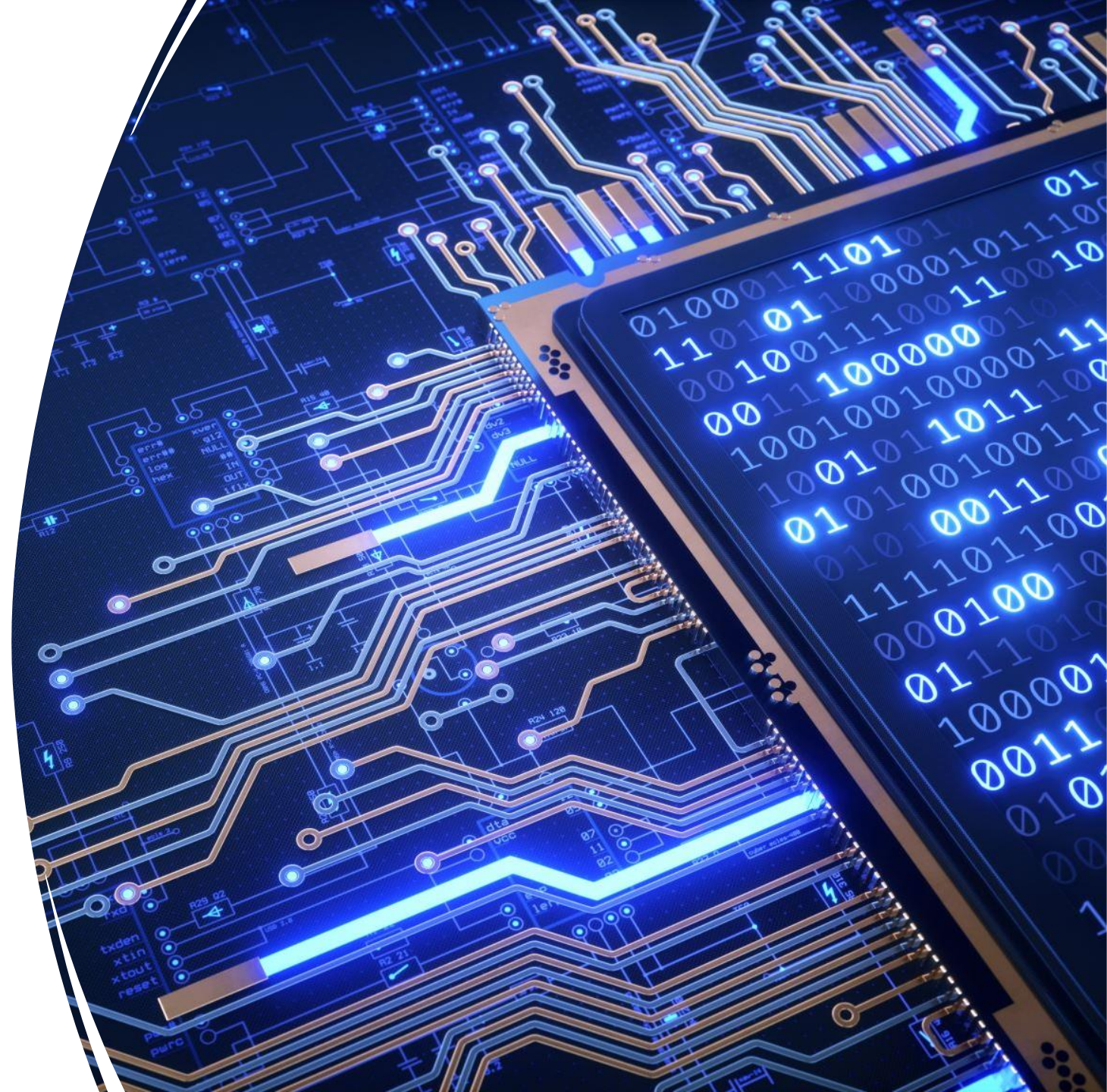




# Implementera Ansible

---

- Efter att Terraform har skapat Compute Engine-instansen kan du använda Ansible för att ytterligare konfigurera denna instans. Detta kan inkludera installation och konfiguration av mjukvarukomponenter och applikationer som behövs på servern.





# Terraform för att sätta upp tjänster

---

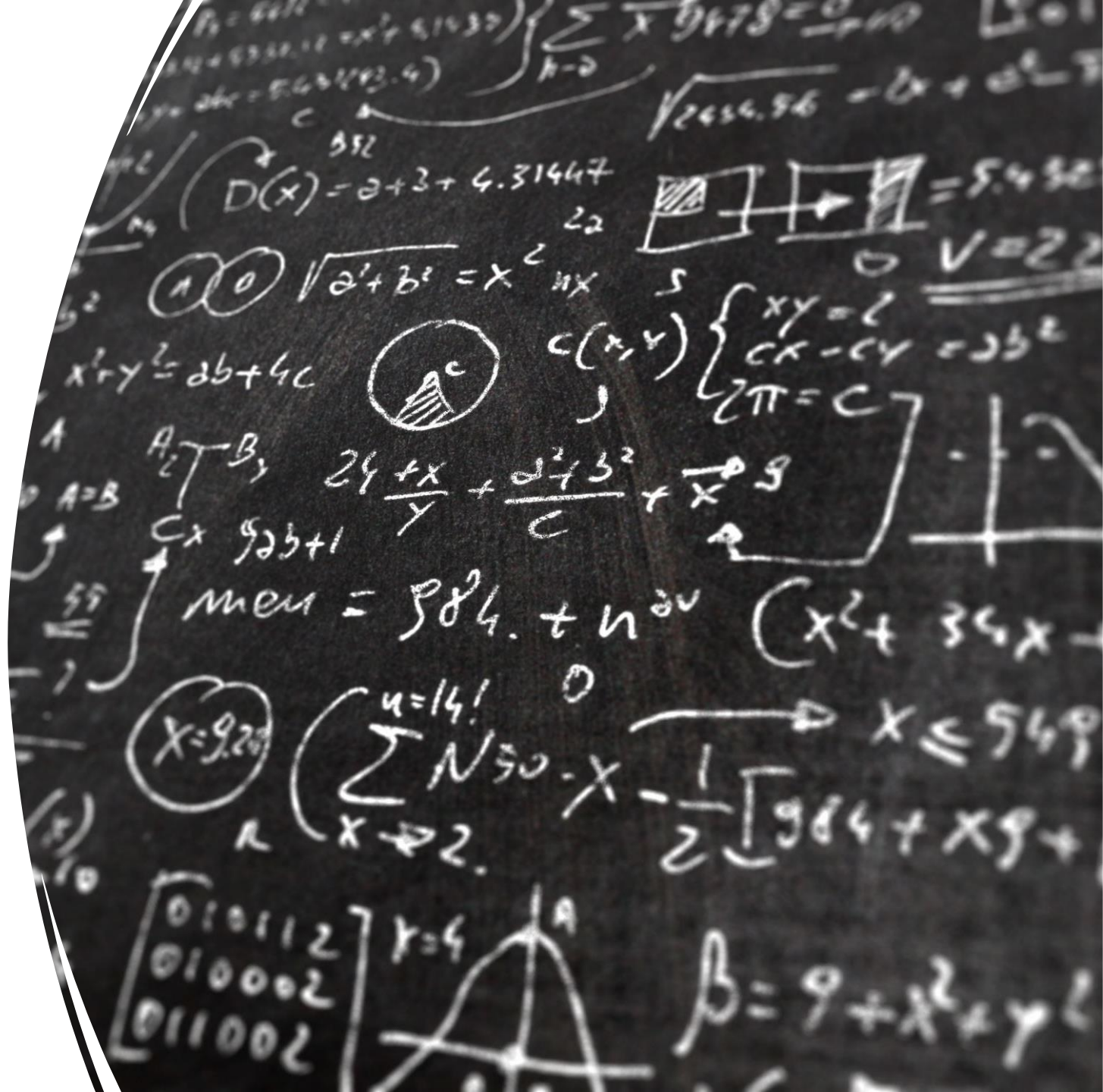
- Terraform används också för att konfigurera de nödvändiga tjänsterna på staging-servern. Detta inkluderar att sätta upp servern enligt behov.





# Databasbackup

- Ett skript används för att hämta en databasbackup från en GCP-bucket och ladda in den i MySQL på staging-servern. Detta är viktigt för att säkerställa att testdata är korrekt.



# Test och Linting

---

- Skriptet utför tester och linting (till exempel phplint) på koden för att säkerställa att den är korrekt och följer kodningsstandarder.





# Godkännande av Tester och Linting

---

- Skriptet kontrollerar om alla tester och linting passerar. Om de gör det, går processen vidare.



# Deployment till Produktion

---

- GitHub API används för att skapa en PR från staging-branch till main-branch för att utlösa produktionens deployment.





# Code Review

---

- Ett viktigt steg där ändringar granskas och godkänns av andra teammedlemmar för att säkerställa kvalitet och säkerhet.



# Deployment till Produktionsserver med Ansible

---

- Använder Ansible för att distribuera till produktionsservern, inkluderar Ansible-åtgärder som en del av GitHub Action som utlöses efter godkänd kodgranskning. Detta säkerställer att produktionsservern uppdateras korrekt efter granskningen.





# Sekvensdiagram

- Hela pipelineprocessen.

