# Software Deployment Process

Ratsameetip Wita Sep 7, 2023

#### About Me

Ratsameetip Wita



AWS Academy Graduate - AWS
Academy Machine Learning

Amazon Web Services Training and Certification





AWS Academy Educator

Amazon Web Services Training and Certification



AWS Academy Graduate - AWS Academy Introduction to Cloud Semester 1

Amazon Web Services Training and Certification



AWS Academy Graduate - AWS Academy Cloud Foundations

2

## Deployment Planning Process

Successful deployment Implementation planning is the result of Prototypes Functional tests Stress tests careful preparation, Deployment Design Deployment architecture Project approva analysis, and design Logical Design ogical architecture Deployment scenario through a series of phases. Usage analysis Use cases Business requirements Business constraints

https://docs.oracle.com/cd/E19199-01/817-5759/intro.html

## Deployment Design Phase

- Design the Deployment Architecture
  - a mapping of the deployment scenario to a physical environment
  - the actual hardware needed to fulfill system requirements and determining a strategy to optimize the deployment architecture to meet budget considerations.

Л

## Deployment Design Phase

- Deployment architecture design directly affects an application's performance, reliability, and scalability.
- It requires careful planning and consideration of the specific needs and goals of the application and its users.

Deployment Architecture

- Servers and Infrastructure
  - Physical or virtual servers, network devices, storage solutions
  - Cloud-based infrastructure
- Application Components
  - Web servers, application servers, databases, load balancers, and microservices.

5

6

## Deployment Architecture [2]

- Scalability and Load Balancing
  - How the application can scale to handle varying levels of traffic or load.
- Data Storage and Databases
  - Decisions about data storage, including databases (SQL, NoSQL), caching mechanisms, and file storage.

## Deployment Architecture [3]

- Security
  - Security measures, such as firewalls, intrusion detection systems, encryption, and access control
- Networking
  - Configuring network settings, including routing, subnets, and firewall rules

## Deployment Architecture [4]

- Deployment Models
  - Depending on the requirements, deployment architectures can be classified into various models like monolithic, microservices, serverless, and container-based.
- Disaster Recovery and High Availability
  - Plans for disaster recovery and high availability should be part of the deployment architecture to ensure minimal downtime and data loss in case of failures.

Deployment Architecture [5]

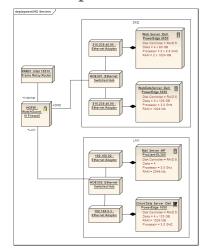
- Monitoring and Logging
  - Implementing monitoring tools and logging mechanisms helps in tracking the application's performance, identifying issues, and troubleshooting problems.
- Maintenance and Updates
  - Strategies for application updates, patch management, and version control are essential to keep the deployed system up to date and secure.

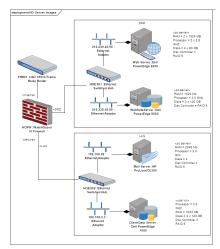
9

## Deployment Diagram

- A deployment diagram is a diagram that shows the physical deployment of a system's components.
- It shows how and where the system is to be deployed
- Benefits
  - Visualize the physical architecture of a system.
  - Identify potential performance bottlenecks or other issues.
  - Used to plan and deploy software systems.
  - Used to document the deployment of a system.

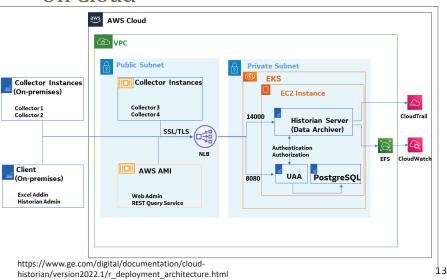
## Deployment Architecture Diagram -On premise



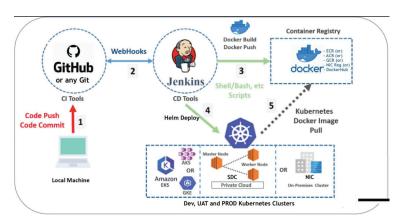


https://sparxsystems.com/enterprise\_architect\_user\_guide/16.1/modeling\_languages/deploymentdiagram.html

### Deployment Architecture Diagram-On Cloud



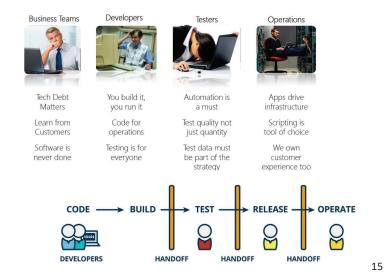
## Deployment Architecture Diagram - Container



https://core.digit.org/platform/architecture/deployment-architecture

14

## Software Development Life Cycle - SDLC



## Deployment Goals and Strategies



https://amazic.com/choosing-the-right-application-deployment-strategy/

## Goals and guidelines

- When uptime is critical, think of minimizing downtime
- Think of rollbacks in case of errors upon deployment or during (stability or performance) tests.
- Put everything (all scripts and configuration) in version control so you can use CI/CD pipelines to deploy new versions of your application quickly and you can track and trace them.
- Deployments should be consistent and repeated across environments to offer smooth promotion of an application from DEV to TEST to ACC to PROD environment.
- Have you thought of backwards compatibility? For example: serve one database to which multiple versions of an application are connected.
- Carry out a root cause analysis (RCA) in case of a deployment failure.

17

## **Application Deployment Strategy**

Strategy	Procedure	Tech Advantage	Business Advantage	Disadvantage
In-Place Upgrade	Replace Version X with Version Y	Simple, Cost- Effective	Cost-Effective for Older Applications	Downtime, Limited Scalability
Blue/Green Deployments	Deploy New Version Alongside Previous Version	Parallel Testing, Minimal Downtime	Zero Downtime, Continuous Operation	Costly, Time- Consuming
Rolling Updates	Replace Instances One by One	Zero Downtime, Parallelism	Zero Downtime, Reduced Costs	Lack of Traffic Control, Handling Multiple APIs
A/B Testing	Serve Different Versions to Test	Full Control, Easy Rollback	Rapid Experimentation, Customer Behavior Insights	
Canary Deployments	Gradually Increase Traffic to New Version	Gradual Testing, Quick Reversion	Fast Feature Testing, Feedback Collection	Not Fail-Safe
Shadow Deployments	Test Production Load on New Feature	No Dedicated Load Test Environment, Mimics Actual Traffic	Realistic Load Testing, Small Increments, Fail Fast	Handling Data Challenges

## DevOps Concept



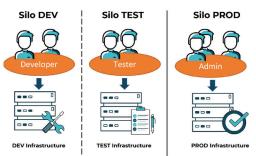
#### 

#### SOFTWARE DEPLOYMENT BEST PRACTICES



### IT silos

 Developers and admins do not speak the same language. Different tasks are prioritized differently and often different work cultures and ways of working clash.



https://www.amitego.com/en/2022/04/25/devops-technical-segregation-of-duties/



22

## What is DevOps

- DevOps is a set of practices that combines software development (Dev) and IT operations (Ops).
- It aims to shorten the systems development life cycle and provide continuous delivery with high software quality.

## The goals of DevOps

 Enhanced collaboration, communication, integration, and automation among developers and IT operations.

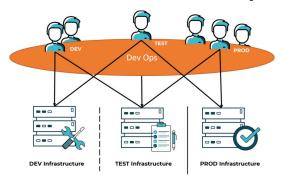


## Key benefits

- Faster time to market
- Reduced failure rate of new releases
- Improved recovery time
- More time to add value instead of fixing/ maintaining.

DevOps strategies

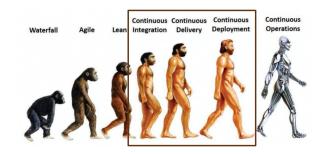
- Developers and admins are now talking mor.
- They are a team and ideally understand which requirements should be considered beyond their own.



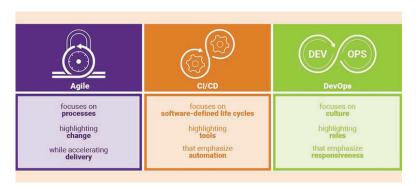
26

## Development of SDLC

- Shorten time on testing and deployment
  - filling the gap in SDLC to DevOps



## Agile, CI/CD and DevOps















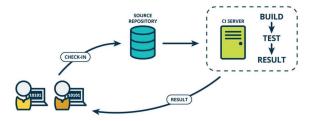




27

## Continuous Integration (CI)

 A development practice that requires developers to integrate code into a shared repository several times a day.



29

## Continuous Delivery (CD<sub>1</sub>)

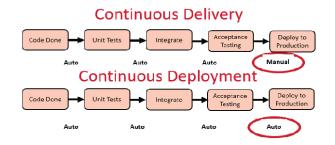
 A software engineering approach where teams produce software in short cycles, ensuring that it can be reliably released at any time.

• The actual deployment to a production environment is a manual process.

30

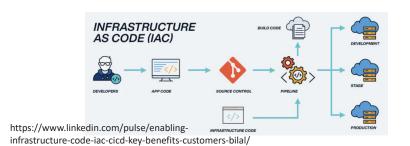
## Continuous Deployment (CD<sub>2</sub>)

 A software release process that uses automated testing to validate if changes to a codebase are correct and stable for immediate autonomous deployment to a production environment.



## Infrastructure as Code (IaC)

 laC is a key DevOps practice involving managing and provisioning infrastructure using machinereadable definition files instead of manual configuration.



## Infrastructure as Code (IaC)

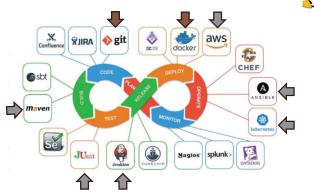
- benefits of IaC: Speed, consistency, minimization of human error, scalability.
- laC is aiding in seamless development, testing, and deployment workflows in CI/CD pipelines.
- laC can be implemented using tools like Ansible, Chef, Puppet, and Terraform.

#### Toolsets to strengthen a DevOps Pipeline

• Which one do you already know?

• bash and Linux skill will be beneficial





33

34

#### Toolsets to strengthen a DevOps Pipeline



## Top 5 DevOps Trends in 2023

- Increased use of Al and ML
- ML algorithms can analyze code changes, predict conflicts, and provide recommendations for improved code quality.
- Al-powered deployment tools can identify and prevent deployment failures before they occur.
- ML algorithms can monitor application performance, detect anomalies, and improve the user experience

https://devopscube.com/devops-trends/

## Top 5 DevOps Trends in 2023

- Expansion of DevSecOps
  - As cyber threats evolve, organizations increasingly prioritize security and compliance.
  - provides a comprehensive and automated security approach that can help detect and remediate vulnerabilities early in development.

https://devopscube.com/devops-trends/

37

39

## Top 5 DevOps Trends in 2023

- Cloud-native Technologies
  - Kubernetes, Docker, and serverless computing are the most popular technologies driving this trend
  - These technologies enable DevOps teams to build and deploy applications with greater speed, efficiency, and agility.

https://devopscube.com/devops-trends/

38

## Top 5 DevOps Trends in 2023

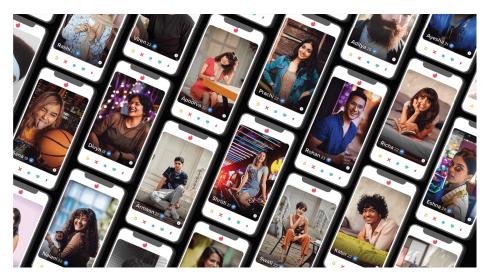
- DevOps as a Service (DaaS)
  - DaaS allows companies to outsource the management of their DevOps infrastructure and processes, reducing the operational burden on their internal teams.
  - DaaS provides a solution to this problem by offering expert DevOps services to organizations that want to outsource their DevOps management.
    - Continuous Integration and Delivery (CI/CD), GitHub Actions, cloud infrastructure management, containerization, monitoring, and security

https://devopscube.com/devops-trends/

## Top 5 DevOps Trends in 2023

- Increased focus on observability
  - Ability to measure and monitor the performance and behavior of applications in real time.
- Distributed tracing enables DevOps teams to trace a request as it travels through a distributed system to identify bottlenecks and other issues impacting performance.
- Log analysis identify patterns and anomalies in order to pinpoint the root cause of problems and make targeted improvements to the application.

https://devopscube.com/devops-trends/





System Design-Tinder | Cost to develop | How to earn revenue

https://dev.to/mukulalpha/system-design-tinder-cost-to-develop-how-to-earn-revenue-4mno

41

## **Dating Application**



- Modern generation of young people embrace the dating lifestyle, the online dating segment continues to grow.
- Tinder is a good example of how dating apps can be transformed into a highly profitable business.
  - high matching conversion rate
  - user-friendly features

42

## Basic Functional Requirements tinder



- Login by multiple ways such as Facebook, Instagram, google, phone number and etc: allow you to sign-up using your Facebook ID, Instagram ID, and others.
- Every user must have their own user profile which includes age, distance, mutual friends, mutual interests, and target description
- The geolocation detect the users' current location
- Users should be able to view recommendations in a geographical region

## Basic Functional Requirements tinder



- Search by filtering (include age, sex, distance, and so on)
- Swipe (Unique value proposition) is an action that shows the user target categories. Right mean like someone, and left mean reject someone
- Match: both users can start to chat once both the users swipe right to each other
- Chat: private message with another user
- Push notification: Notify the user when getting a new message like or match or chat

#### Scale



- 57+ million members
- 12+ million active users
- 55+ Billion matches
- 100+ Million downloads
- 1.8 billion swipes every day (left swipe, right swipe, super like)
- 40+ Language support

Deployment Architecture for



45

46

## Deployment Architecture **tinder**

Tinder is completely hosted on **AWS cloud**. It doesn't have any web application but IOS and Andriod. Tinder uses **AWS amplify** to build and test mobile applications, MongoDB for DB, and Redis for caching and in-memory database.

## **Tinder Architecture**



High Level Design



Fig: Tinder Architecture

#### Tinder Architecture



- A fleet of microservices will be behind the Gateway to serve user requests.
- The Profile Creator Service will be invoked when the user profile is created.
- The user will be added to the corresponding geo-sharded index so that the user shows up in recommendations of nearby users.
- The Recommendation Service will query this index when it receives a request to generate recommendations for other users.
- Once the user starts swiping through the recommendations, the Swipes Service will receive those swipes and place them in a data stream (e.g., AWS Kinesis/SQS).
- A fleet of workers will read data from those streams to generate matches.
- The workers do this by querying the LikesCache to determine if it's a match.
- If it is a match, the match notification will be sent to both users using technologies such as WebSockets.

49

51

### Recommendation Engine **Properties**



- Low latency: When a person logs in to the application, we need to load profiles/potential matches profiles real quickly.
- Not realtime: It's okay if it's not realtime i.e if someone newly joins tinder it's okay if it take a min to show this person's profile on our account.
- Easy to shard/distribute: Since we have tons of profiles from across the globe, this recommendation engine should be able to shard the data as we can't keep it in one system.
- Full-text search: we need to search thorugh the whole profile of an individual, to provide better recommendations
- HTTP interface: or web socket to get the data and send it to the application.
- Structure data: XML/JSON

50

52

## Recommendation Strategy **tinder**



- Tinder basically wants people to meet.
  - If I am a user from location X, India, I will obviously like to get a match with someone who is from location X + 50km.
- How to make low latency in elastic search?
  - Shard the data by geographical location.!!
  - ...how?

It's a Match!



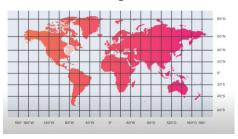


Elasticsearch is able to achieve fast search responses because, instead of searching the text directly, it searches an index instead

## Geosharding



- Requests originating from a box are served by the server in that box.
- Boxes with high population densities may need more than one server.
- The size of the boxes in different areas is determined by Unique user count, active user count and query count from these regions.





## Geosharding



- When a person opens Tinder, their phone makes a query to a mapper system.
- The mapper system uses the user's latitude and longitude to determine which server their data is stored on.
- This server is the same server that stores the user's potential matches.
- The mapper system can be located in any physical location.
- All of the data for a particular cell will reside on that one server.

54

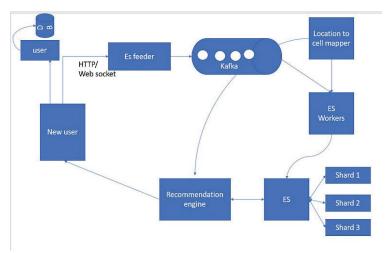
## Recommendation on Geosharding



- The information for cells 1, 2, 3, 4, 5, 6, and 7 is stored on servers 1, 2, 3, 4, 5, 6, and 7, respectively.
- You are a Tinder user residing in cell 3 and have set your range to 100 km.
- This means that you want to know all your potential matches within 100 km of your location.
- Your information resides on server 3.
- The potential matches' information resides in the radius of 100 km, which includes cells 1 to 7.
- When you open Tinder, your request will be sent to all the servers, i.e., servers 1 to 7.
- The servers will then gather the recommendations and send them back to you.

# Recommendation on Geosharding





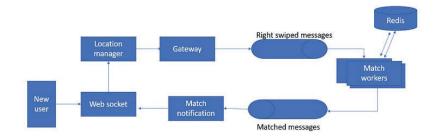
## Matching



- There are millions of matches happening every day.
- We can have one matchmaking service per cell or group a few cells together with one matchmaking service.
- There will be a few matchmaking services running at the same time to balance out the queries per location.
- Each matchmaking service will belong to a few cells instead of just one cell.
- Matches will not happen between countries, they will happen in the cell where a profile is recommended to a user.
- For example, if we recommend 100 profiles to a user, there is a good chance that only 20-30 of them will be swiped right.

Matching





57

## Matching



- When a user swipes right, a message is sent to the matchmaking service.
- The location manager determines which shard or matchmaking service this message should go to.
- The message is then redirected to the gateway, which connects to Kafka.
- The message is now in the gueue.
- Depending on the number of shards, there will be one or more matchmaking services that this information will be broadcasted to.
- The information captured here is who swiped right on whom, their location, and other metadata.
- There can be parallel workers that keep reading messages from the Kafka queue.

## Matching [2]



- If A swipes right on B, an entry like "A\_B" is added to Redis.
- When B swipes right on A, the same process happens.
- The match worker picks up the message and checks Redis to see if A has ever swiped right on B.
- If they have, this means that a match has happened and a message is added to the matched queue.
- The match notification then picks up this message and sends it to both A and B through WebSockets.

## More reading on **tinder**

- https://interviewnoodle.com/tinder-systemarchitecture-2012902cb415
- https://medium.com/system-design-concepts/datingapplication-system-design-aae411412267
- https://www.techaheadcorp.com/blog/understandingsystem-design-architecture-of-tinder/
- http://highscalability.com/blog/2022/1/17/designingtinder.html

### References

- https://www.jitendrazaa.com/blog/salesforce/continuo us-integration-vs-continuous-delivery-vs-continuousdeployment/
- https://devopscube.com/devops-trends/
- https://dev.to/pavanbelagatti/learn-how-to-setup-acicd-pipeline-from-scratch-for-a-go-application-4m69
- https://www.hostinger.com/tutorials/ssh/basic-sshcommands
- https://www.hostinger.com/tutorials/ssh-tutorial-howdoes-ssh-work