



A Practical Evaluation of Deployment Strategies for Services Running in the Cloud

Vlad-Stefan Dieaconu

*National University of Science and
Technology Politehnica Bucharest*
vladstefandieaconu@gmail.com

Alexandru Bobe

Ovidius University of Constanta
alexandru.bobe@univ-ovidius.ro

Iosif-Alexandru Selea

CrowdStrike
alexselea@gmail.com

Motivation | Why is this important?

01 Bad deployments cause real-world impact

2025 Google Cloud outage

→ 1.4M users, 50+ services down,
7h recovery

Knight Capital (2012)

→ \$440M lost in 45 minutes

02 Cloud Services are Everywhere

94% of companies rely on cloud services (Edge Delta, 2024)

\$675B+ in 2024 for global cloud market (Gartner)

03 ½ Cloud Incidents

VOID Database, Aldea 2025

50%+ of failures were directly caused by deployment-related changes

04 ++ Robust Deployment Strategies

Internal Microsoft study:

→ **13% of high-severity outages** came from deployments

Google research:

→ **16% of service failures** were caused by deployments

Contents

Software Deployment Strategies

- All-At-Once
- Rolling
- Blue-Green
- Canary
- A/B
- Shadow

Evaluation Framework

- Six key criterias
- Measurement system

Conclusions

- The best deployments strategy?

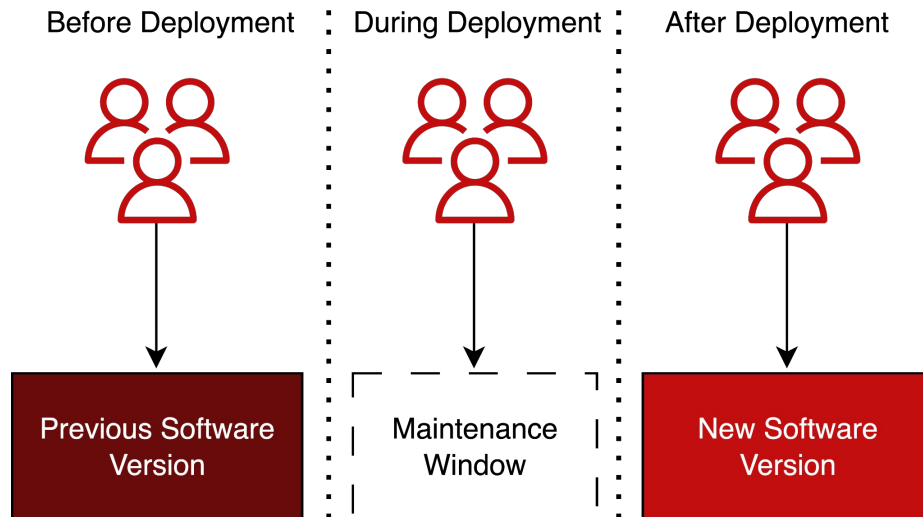
All-at-Once / Big-Bang Deployments

Like renovating your entire house while you're still living in it

- Fast & simple
- High downtime
- Need for a maintenance window
- Rollback = redeploy everything

Possible implementations:

- **In-Place:** updates existing infrastructure
- **Recreate:** terminate old & create new infrastructure



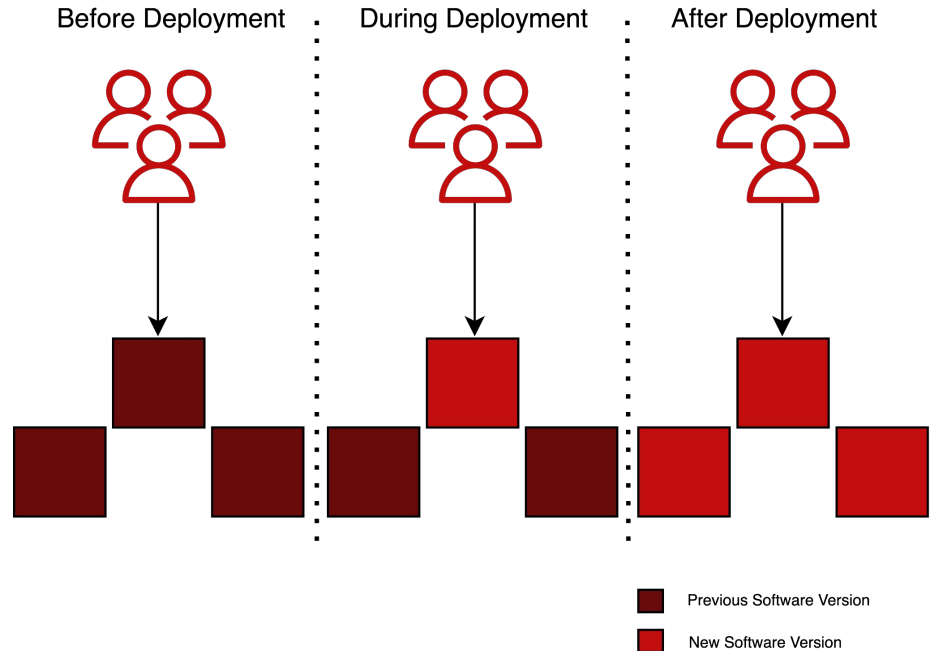
Rolling Deployments

Like renovating your house one room at a time — you can still live there, but some rooms might be unavailable

- Update servers gradually
- Zero/minimal downtime
- Requires backward compatibility
- Rollback = redeploy everything

Possible implementations:

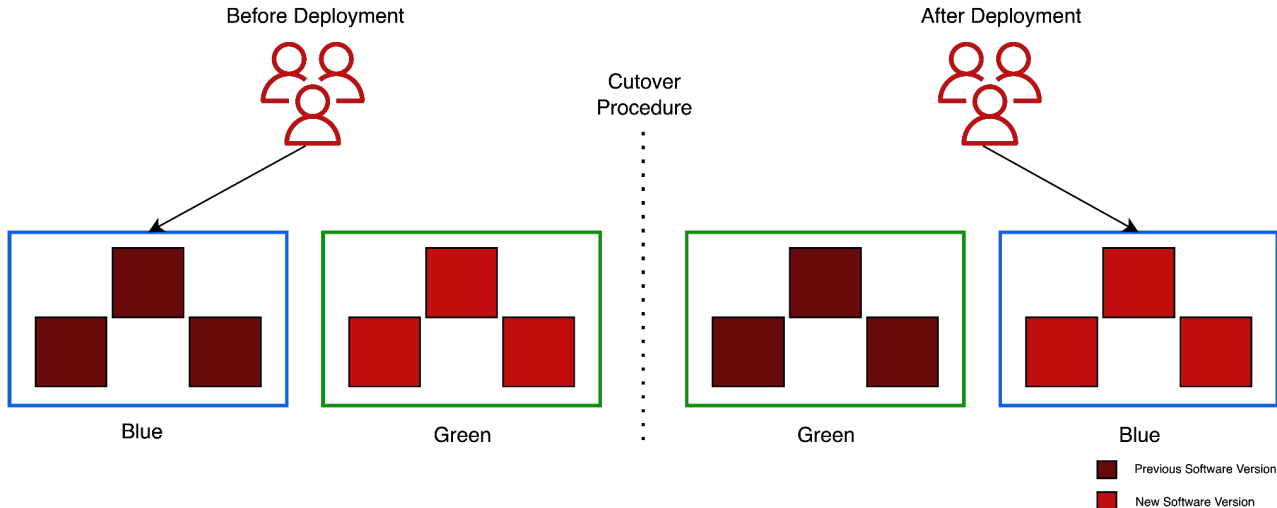
- **In-Place** Rolling Deployments
- **Immutable** Deployments



Blue-Green Deployments

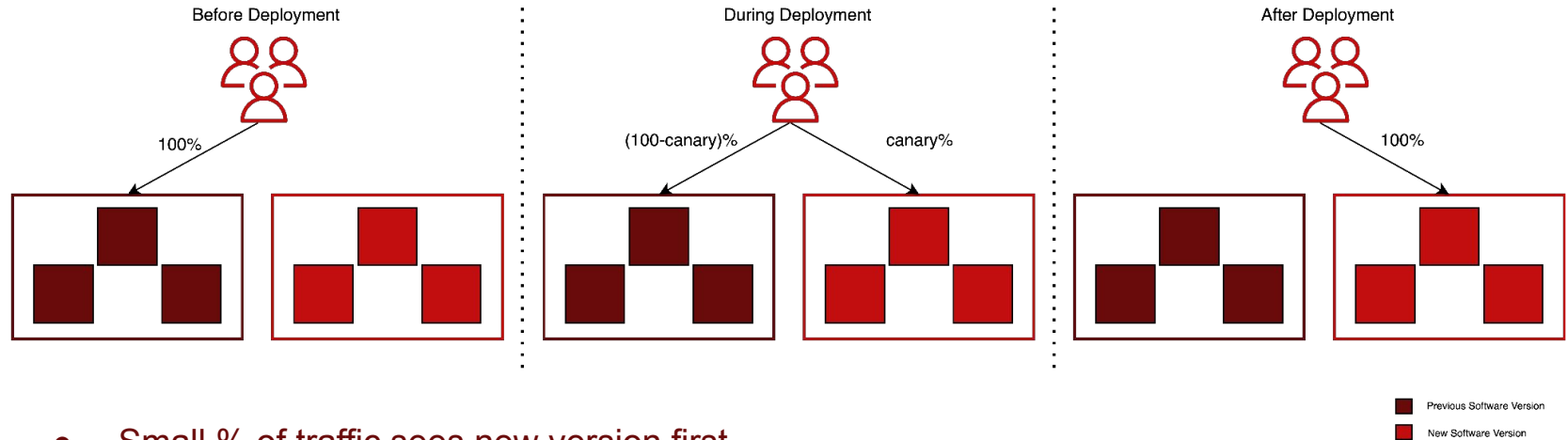
Like having an identical twin of your entire house. You live in one, set up the other, then switch when ready. If something's wrong, just move back.

- Two identical environments
- Instant switch & rollback
- Ability to test on inactive side
- Higher infra cost



Canary Deployments

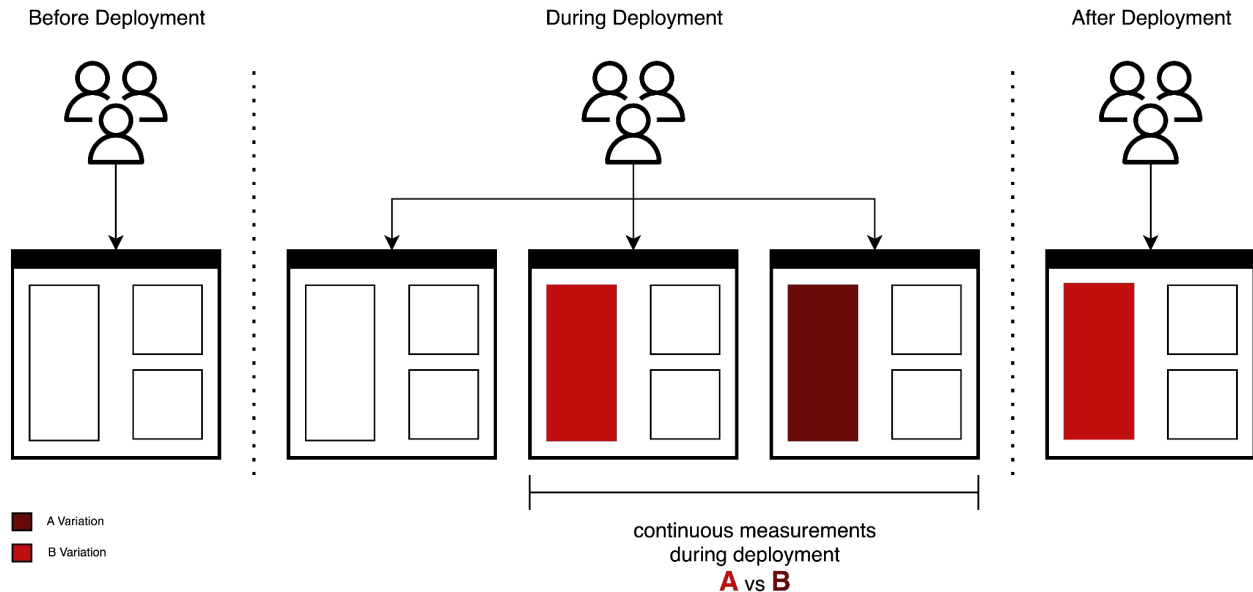
Like taste-testing your cooking on one family member before serving dinner to everyone



- Small % of traffic sees new version first
- Real-world testing, low risk
- Instant rollbacks inherited from Blue-Green
- Needs strong observability

A/B Deployments

Like giving half your guests a blue button and half a red button, and seeing which one they click more

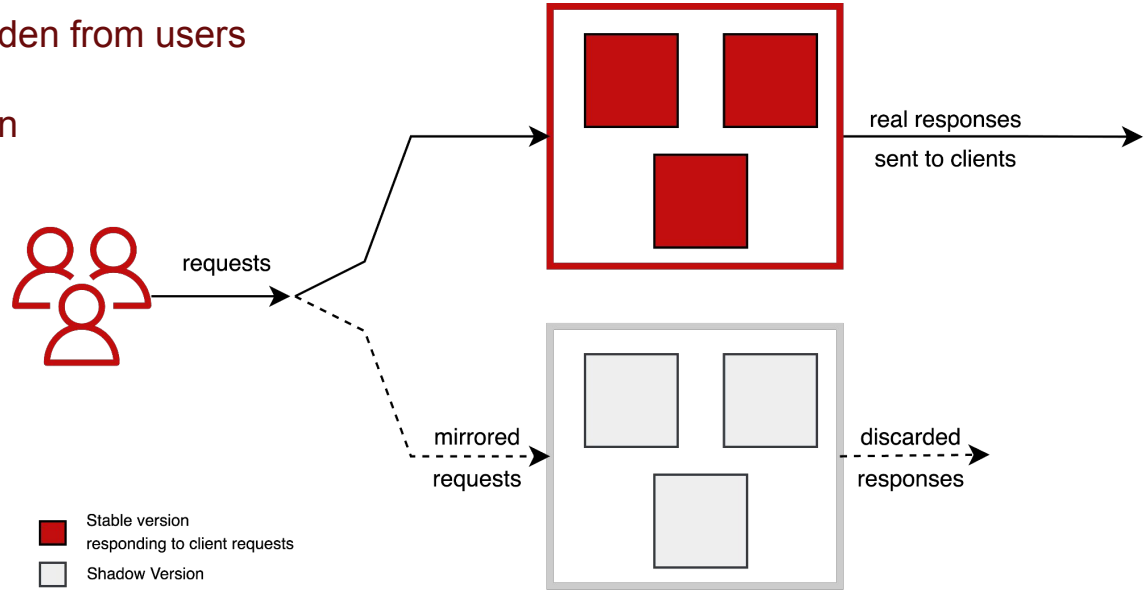


- Multiple versions live at once
- Compare user behavior/metrics
- Focus on optimization, not just stability

Shadow Deployments (Dark Launches)

Like a dress rehearsal where the actors perform but the audience can't see it

- Duplicate environment, hidden from users
- Real traffic tested silently
- Expensive & complex to run

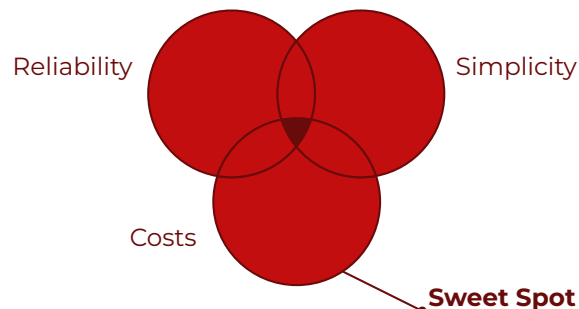


Evaluation Framework

- **1. Deployment Speed**
 - How fast can you get your code live?
- **2. Downtime**
 - Will your users notice anything?
- **3. Risk Level**
 - What's the chance everything goes sideways?
- **4. Rollback Speed**
 - How quickly can you hit the "undo" button?
- **5. Infrastructure Cost**
 - What's this going to cost you?
- **6. Operational Complexity**
 - How much of a headache is this to manage?

Strategy	Speed	Downtime	Risk	Rollback	Cost	Complexity
All-at-Once	L	XL	XL	XS	S	XS
Recreate	M	XL	XL	XS	S	S
Rolling	M	M	L	S	S	XS
Immutable	M	M	L	S	M	S
Blue-Green	XL	M	M	L	L	M
Canary	S	S	S	M	L	L
A/B	S	S	S	XL	XL	XL
Shadow	S	XS	XS	XL	XL	XL

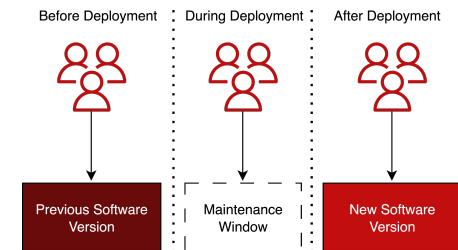
Legend: X=Extra, S=Small, M=Medium, L=Large.



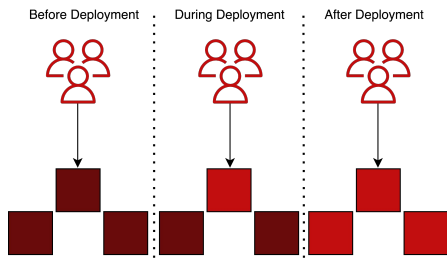
**There's no One-Size-Fits-All
Deployment Strategy**

Thanks

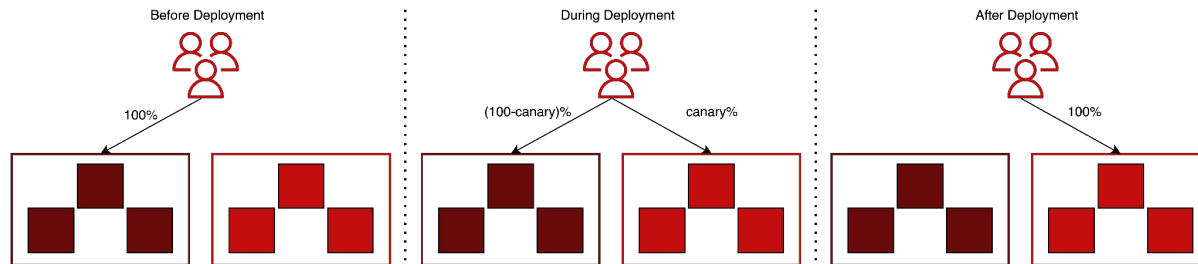
Do you have any questions?
vladstefandieaconu@gmail.com



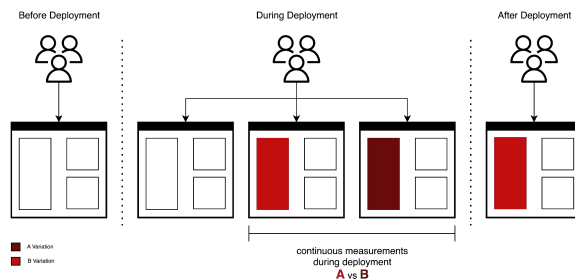
1. Big-Bang Deployments



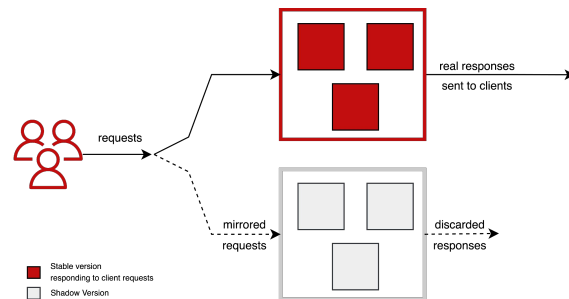
2. Rolling Deployments



4. Canary Deployments



5. A/B Deployments



6. Shadow Deployments

3. Blue-Green Deployments

