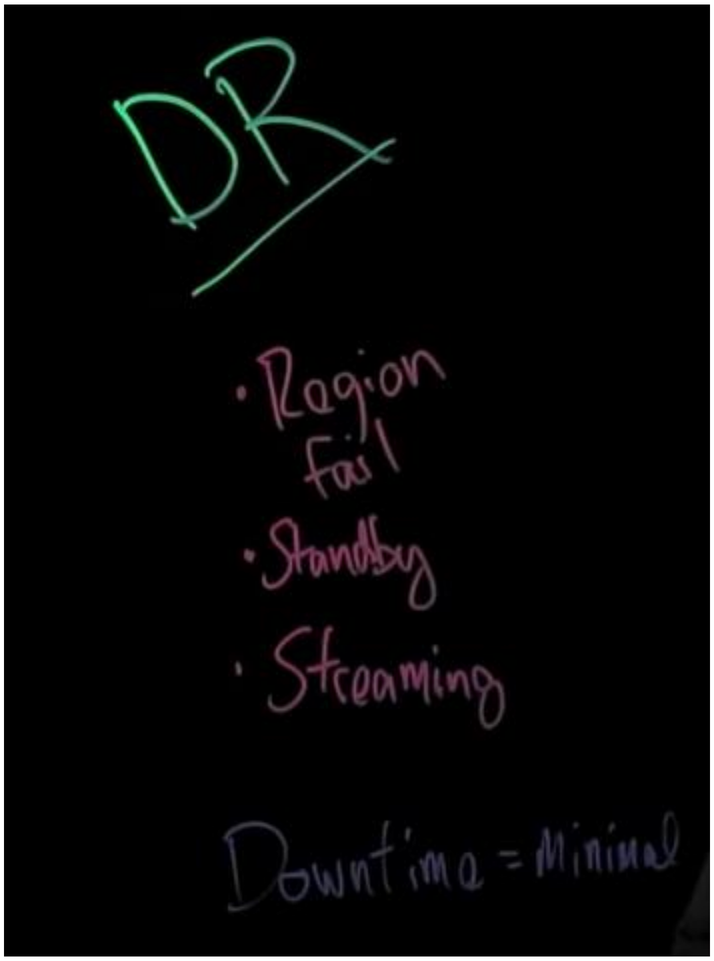
Discovery Recovery and Back Up (Making Snapshots)

- The biggest challenge to successful DevOps backups stems from the fact that it's a **live operational environment.**
- DevOps environments are rich with highly dynamic microservices that run across a highly distributed system of tightly connected compute, data, applications, and services.
- As applications and services move to a multi-region and multi-tenancy architecture, it's critical to consider two types of backups: persistent data backup and system state backup.
- Persistent backups deal with data storage devices that retain data when power is off, including databases, static resources, and messages.
- System state backups back up operating system files and include cluster state, node state, microservice state and data pipeline state.

What type of data back up should we pursue as DevOps Engineer?

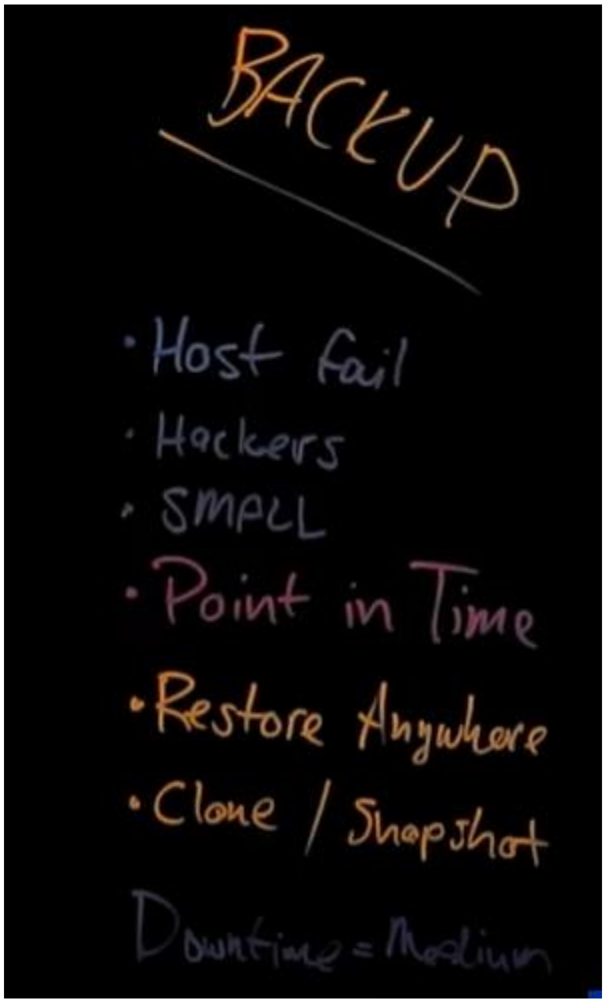
Running



DR

- Region fail
- Standby
- Streaming

Downtime = minimal



BACKUP

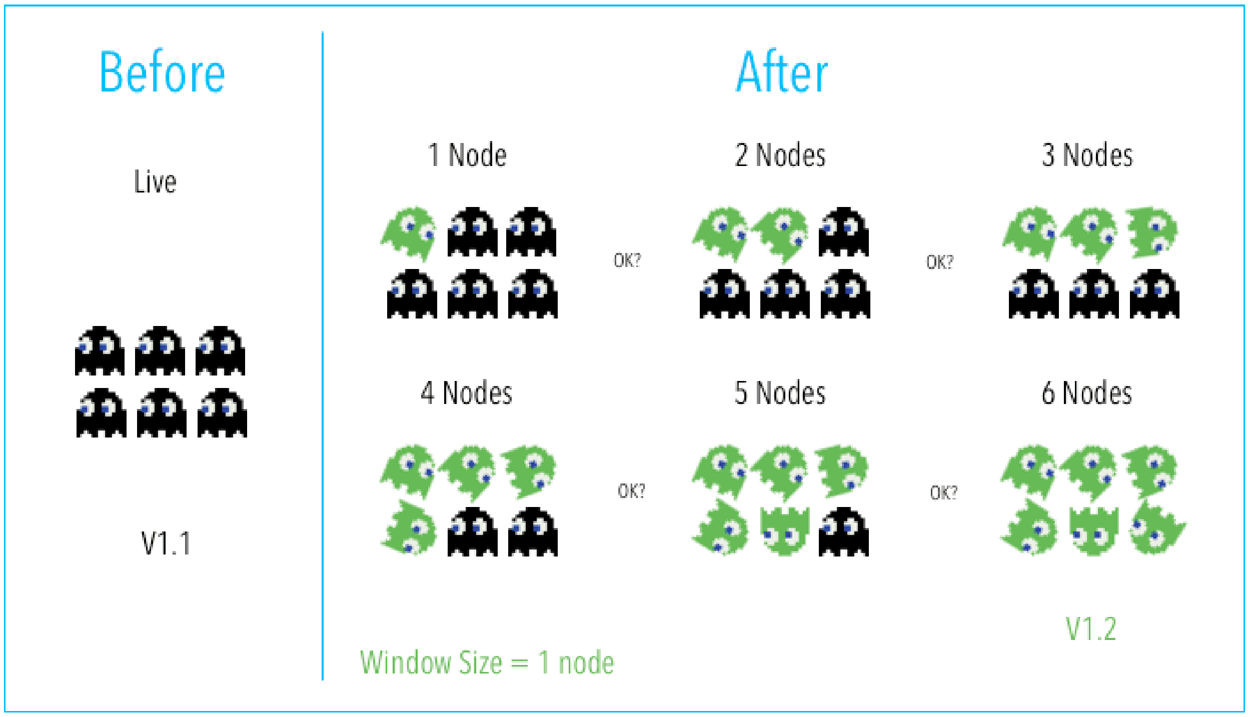
- Host fail
- Hackers
- SMALL
- Point in Time
- Restore Anywhere
- Clone / Snapshot

Downtime = Medium

Think about these deployment strategies

Rolling/ Rolling update Deployment

- A rolling deployment is a deployment strategy that updates running instances of an application with the new release.
- All nodes in a target environment are incrementally updated with the service or artifact version in integer N batches.



Pros:

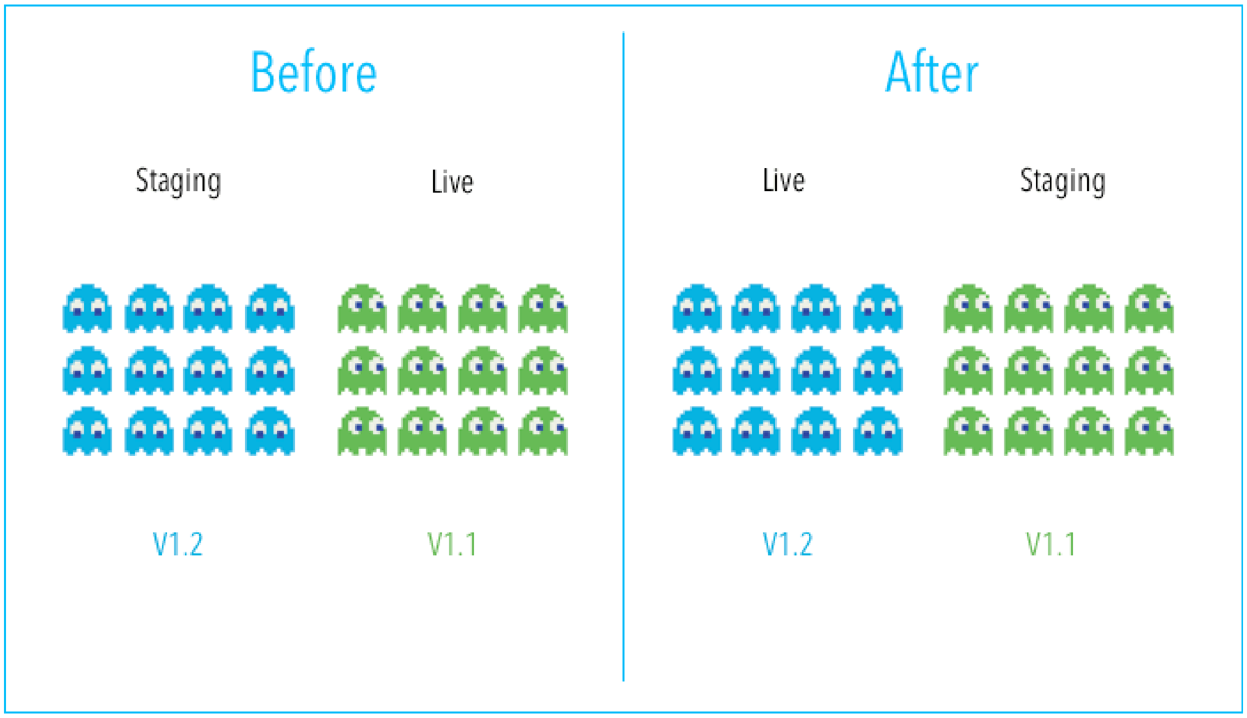
- The benefits of a rolling deployment are that it is relatively simple to roll back, less risky than a basic deployment, and the implementation is simple.

Cons:

- Since nodes are updated in batches, rolling deployments require services to support both new and old versions of an artifact.
- Verification of an application deployment at every incremental change also makes this deployment slow.

Blue-Green Deployment

- Blue-green deployment is a deployment strategy that utilizes two identical environments, a “blue” (aka staging) and a “green” (aka production) environment with different versions of an application or service.
- Quality assurance and user acceptance testing are typically done within the blue environment that hosts new versions or changes.
- User traffic is shifted from the green environment to the blue environment once new changes have been testing and accepted within the blue environment.
- You can then switch to the new environment once the deployment is successful**.**



Pros:

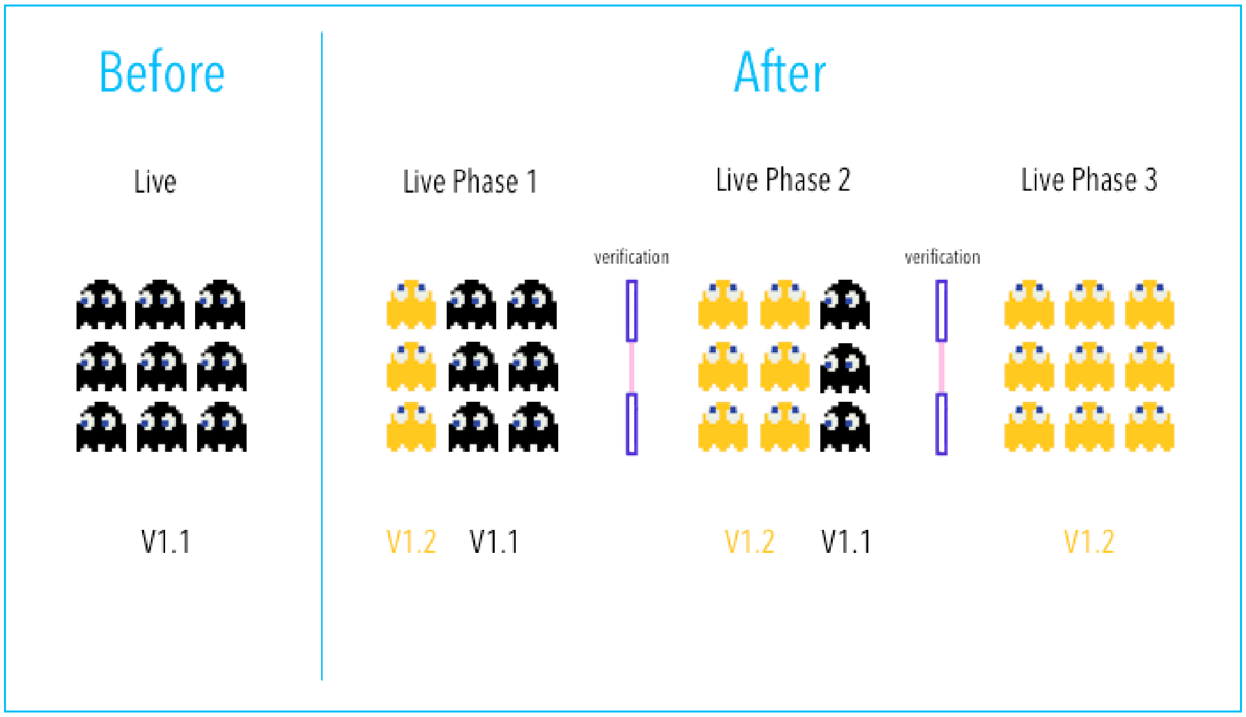
- One of the benefits of the blue-green deployment is that it is simple, fast, well-understood, and easy to implement.
- Rollback is also straightforward, because you can simply flip traffic back to the old environment in case of any issues.
- Blue-green deployments are therefore not as risky compared to other deployment strategies.

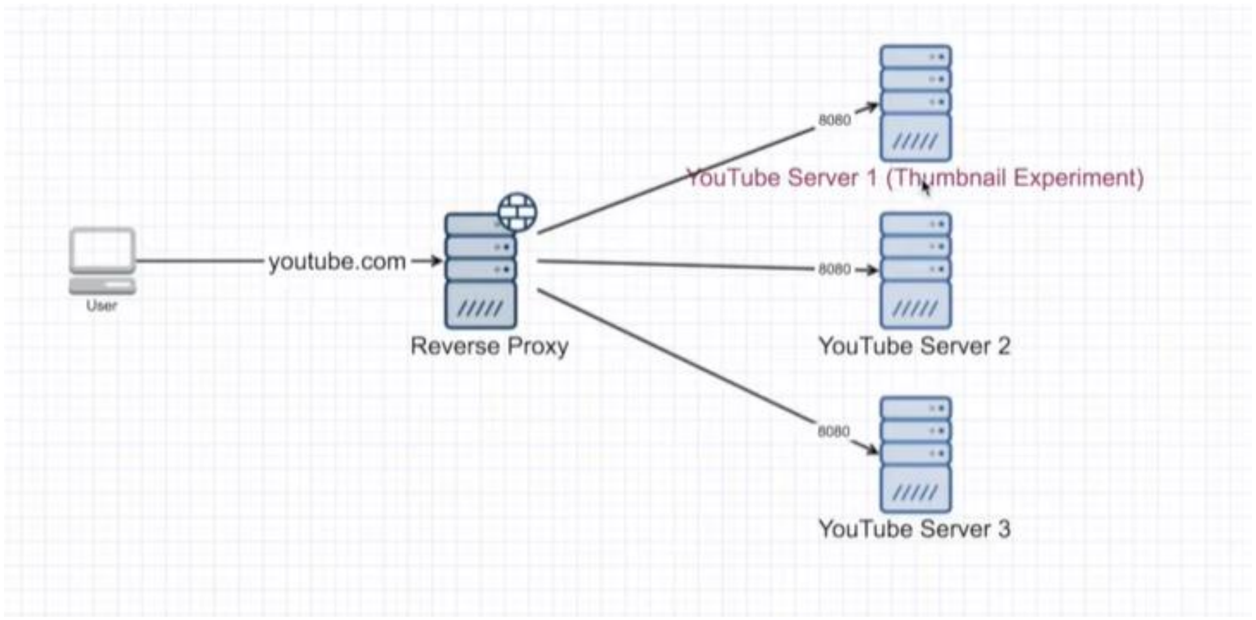
Cons:

- Cost is a drawback to blue-green deployments. Replicating a production environment can be complex and expensive, especially when working with microservices.
- Quality assurance and user acceptance testing may not identify all the anomalies or regressions either, or so shifting all user traffic at once can present risks.
- An outage or issue could also have a wide-scale business impact before a rollback is triggered, and depending on the implementation, in-flight user transactions may be lost when the shift in traffic is made.

Canary Deployment

- A canary deployment is a deployment strategy that releases an application or service incrementally to a subset of users.
- All infrastructure in a target environment is updated in small phases (e.g: 2%, 25%, 75%, 100%).
- A canary release is the lowest risk-prone, compared to all other deployment strategies, because of this control.





The reverse proxy or the load balancer will be configured in a special way to allow 3% of the requests to be sent to the specific server where experiment is being conducted.

What is the difference between reverse proxy and load balancer?

A reverse proxy accepts a request from a client, forwards it to a server that can fulfill it, and returns the server's response to the client.

A load balancer distributes incoming client requests among a group of servers, in each case returning the response from the selected server to the appropriate client.

Pros:

- Canary deployments allow organizations to test in production with real users and use cases and compare different service versions side by side.
- It’s cheaper than a blue-green deployment because it does not require two production environments.
- And finally, it is fast and safe to trigger a rollback to a previous version of an application.

Cons:

- Drawbacks to canary deployments involve testing in production and the implementations needed.
- Scripting a canary release can be complex: manual verification or testing can take time, and the required monitoring and instrumentation for testing in production may involve additional research