**EBS**

EBS 🡺 Elastic Beanstalk (Regional service).

**Platform as a Service.**

It is a fully managed service offered by Amazon Web Services (AWS) that simplifies the deployment and management of applications in various programming languages.

It abstracts away the underlying infrastructure details, allowing developers to focus on writing code rather than dealing with the complexities of configuring and managing servers.

**Beanstalk Deployment Options for Updates :**

**1) All at once :** In this deployment strategy, Elastic Beanstalk updates all instances simultaneously. This approach is quick but may result in downtime during the deployment process. If your application can tolerate a brief period of unavailability, this method may be suitable.

**A diagram of a diagram

Description automatically generated**

**2) Rolling :** Rolling deployments update a specified number of instances at a time, minimizing downtime. During the update, Elastic Beanstalk takes a subset of instances out of service, updates them, and then moves on to the next subset until all instances are updated. This method helps maintain some level of application availability during the deployment process.

A diagram of a bucket size

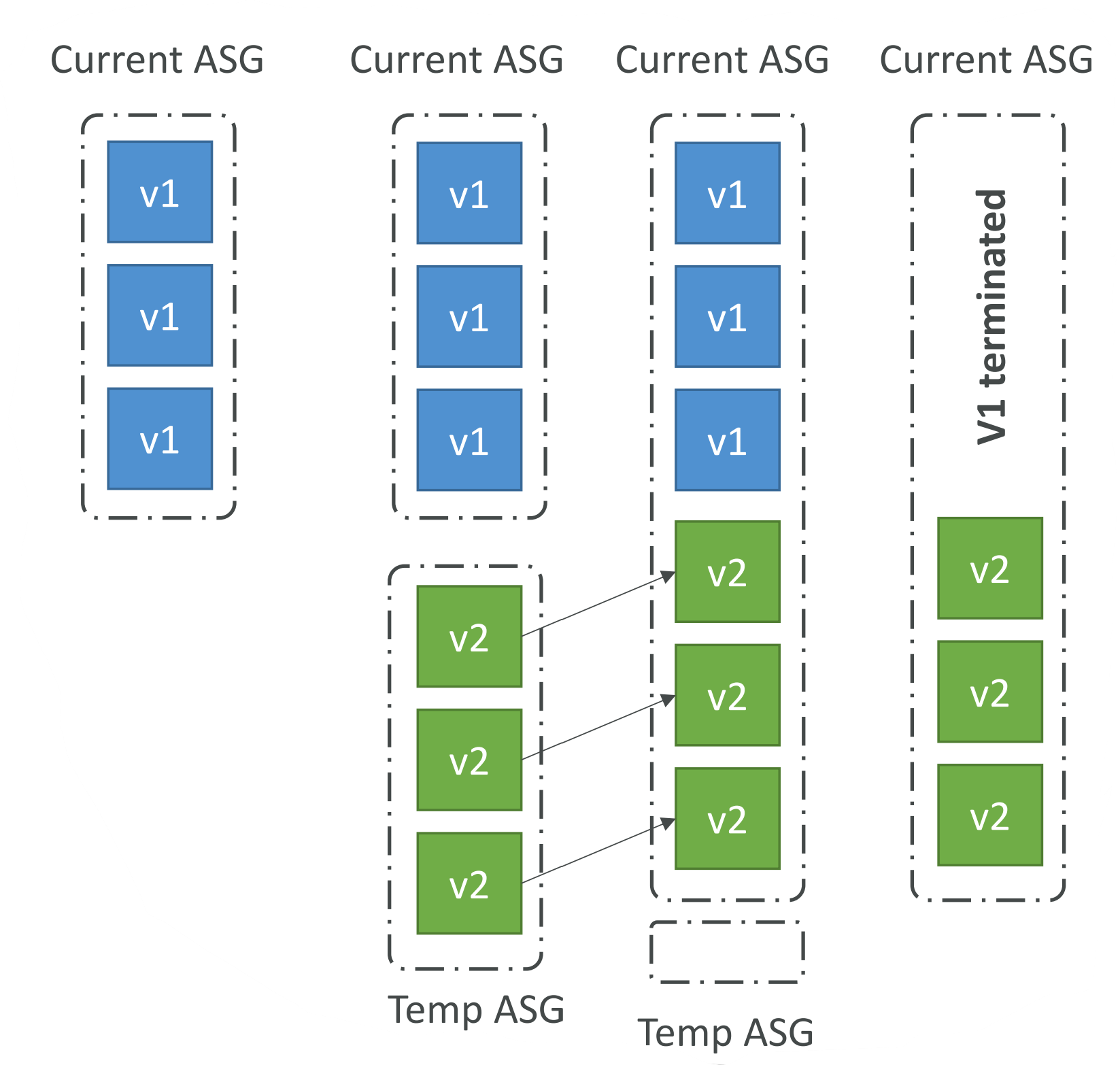
Description automatically generated

**3) Rolling with additional batches :** This strategy is similar to rolling deployments but allows you to keep extra instances running during the update. This can help handle additional traffic or provide extra capacity while the deployment is in progress.

A screenshot of a computer screen

Description automatically generated

**4) Immutable :** Immutable deployments replace instances with a new set of instances rather than updating the existing ones. Elastic Beanstalk creates a new Auto Scaling Group with the new instances and swaps them with the old group when the new instances are ready. This approach minimizes downtime but may require additional resources during the deployment process.

****

**5) Blue-Green :** Blue-Green deployments involve running two separate environments, a "blue" one with the current version and a "green" one with the new version. Traffic is switched from the blue to the green environment after the update is complete. This approach virtually eliminates downtime and allows for easy rollback if issues are detected in the new version.

