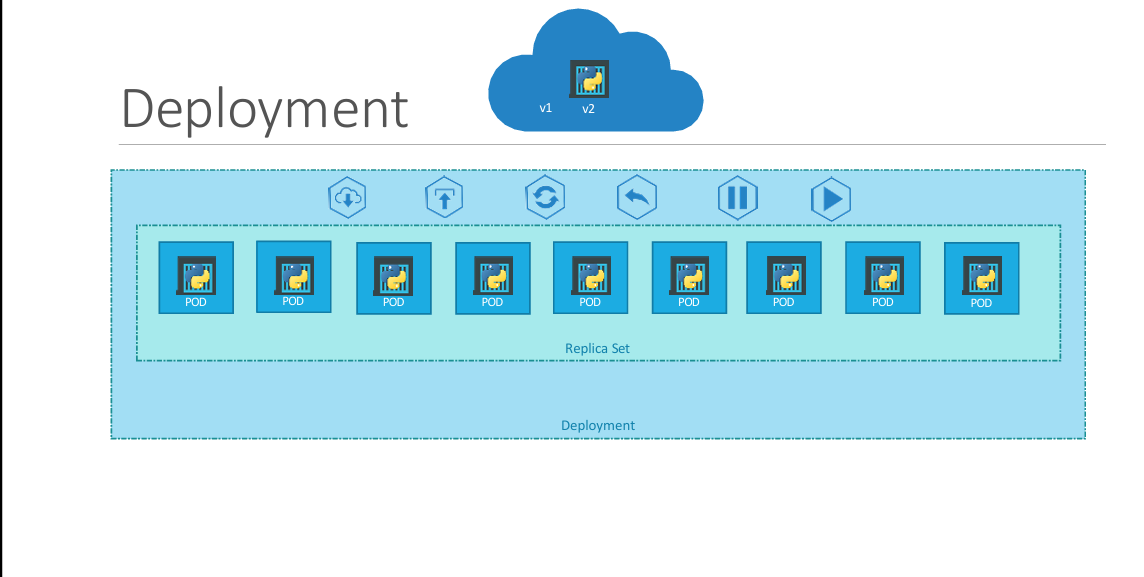
**Deployment**

****

**When upgrading your instances, it’s important to avoid upgrading all of them simultaneously, as this could disrupt users accessing the applications. Instead, a safer approach is to upgrade them sequentially, a process known as *Rolling Updates*.**

**Now, suppose an upgrade introduces an unexpected issue, and you need to revert to the previous version. In such cases, you would perform a *rollback* to undo the recent changes.**

**Additionally, consider a scenario where multiple modifications are required in your environment—such as upgrading the web server version, scaling the infrastructure, or adjusting resource allocations. Rather than applying each change immediately, you may prefer to *pause* the environment, make all necessary updates, and then *resume* to roll them out collectively.**

**Kubernetes Deployments provide all these capabilities, allowing seamless instance upgrades, rollbacks, and controlled rollout of multiple changes.**

**Earlier, we discussed *Pods*, which encapsulate single instances of an application, such as a web service. These Pods are managed using *Replication Controllers* or *ReplicaSets*, ensuring the desired number of instances remain running. At a higher level in Kubernetes architecture, *Deployments* offer more advanced functionalities, including rolling updates, rollback capabilities, and the ability to pause and resume changes, making application management more efficient and resilient.**