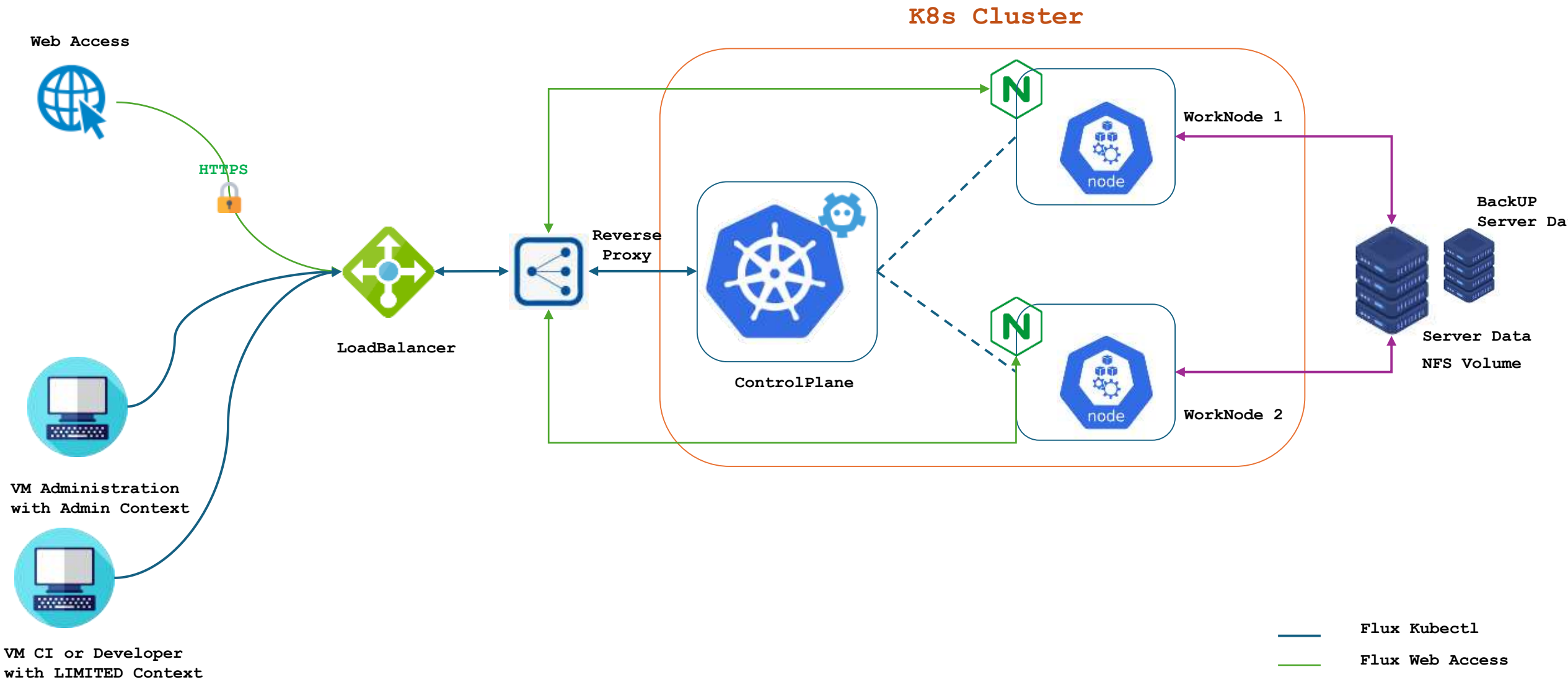
Kubernetes Architecture

K8s Architecture Design



K8s Architecture Design Components Details

- ControlPlane (Master Node)
  - **kube-apiserver**
  - Etcd
  - kube-scheduler
  - kube-controller-manager
- Worker Node
  - Kubelet
  - Kube-proxy
  - Pods
  - Container-runtime engine
  - Container Network
  - Ingress Controller as a **DaemonSet** (not a default component)
    - *Ingress Controller was deployed as a DaemonSet to ensure the High Availability of the Web Access.*
  - Any components of any application will be deployed on the worker Nodes due to the default taint of our controlPlane (by default the scheduler can not deploy any application on the Master Node)
- LoadBalancer
  - We use LB to have an external access to our Cluster and deployed application

- Reverse Proxy
  - It used to manage and secure different access type.
  - For example : we can not have a direct access to our kube-api without passing through the reverse-proxy.
- Server Data
  - We use a dedicated server to persist important data like Databases data which also can facilitate the data backup and Restore workflow, so any application need to persist data must use a NFS volume type.
- VM Administration
  - It’s not mandatory, kubernetes administrator can also use his own server, PC but he need to get the kube-config file which contains the admin context
- VM CI or Developer
  - For CI servers (Jenkins, Gitlab-runners, …) can use kube-config context with specific access to just deploy applications.
  - For Developers they can use their own servers or PC with kube-config file that contains the definition of limited context.
- Web Access
  - Once the application was deployed we can access its UI via the Ingress controller, just an important information Helm charts must contains Ingress resource definition.

Access Workflow With Components communication