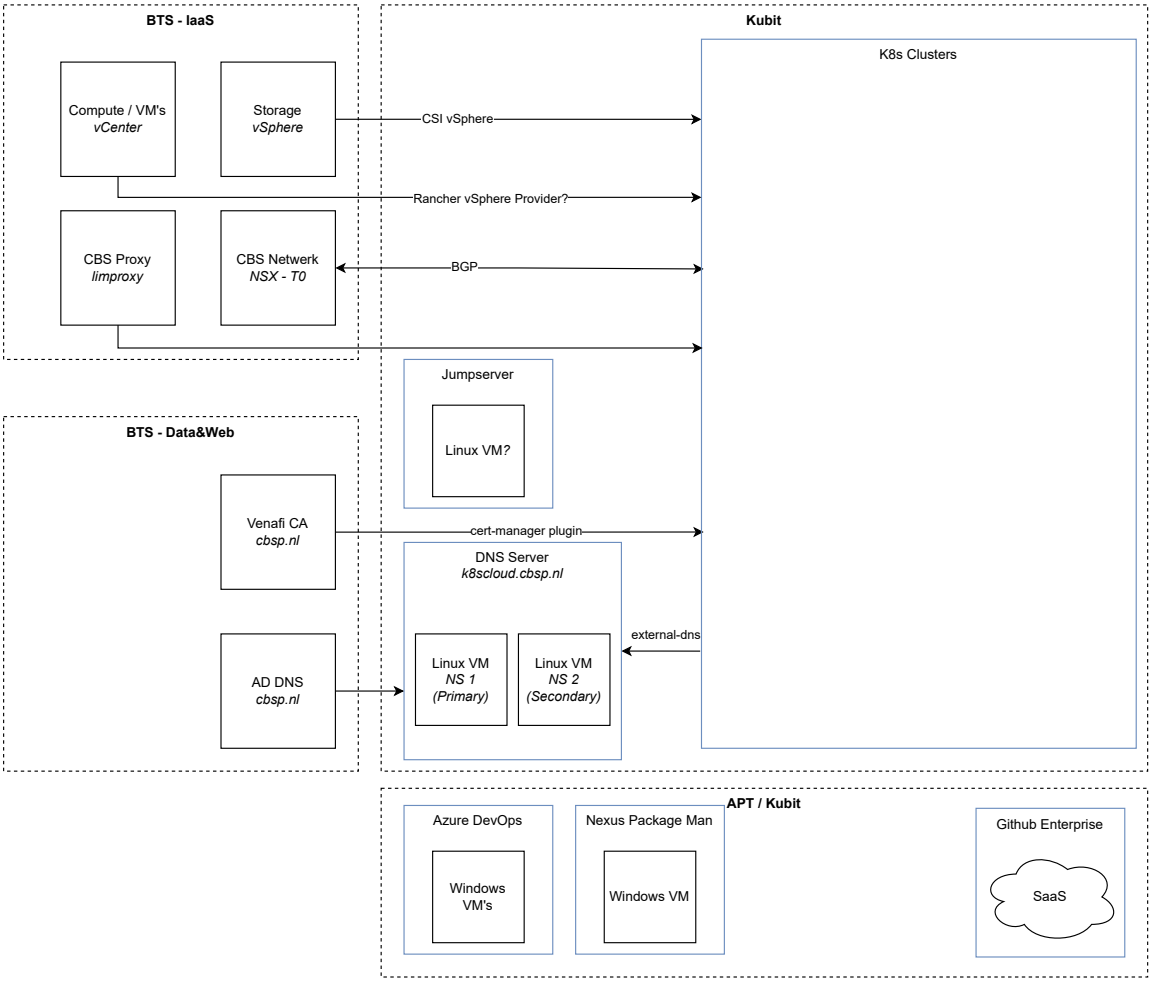
Infra dependencies

For running a k8s cluster on premise, we need to have some infrastructure in place. We need for example computing equipment, storage, networking connectivity and more. K8s is running on top of these parts or needs a tight integration with it.

Required infra components

In the diagram below all the required infrastructure components can be seen, including the CBS department being responsible for it (dotted surrounding area). Each component is being explained in the next paragraphs (below the diagram).



Compute power (CPU & RAM)

K8s needs CPU and RAM to be able to run itself and its components. These parts can be delivered by physical machines, but commonly these parts are delivered via Virtual Machines (VM's). These VM's can easily being spinned up when additional computing power is required. Also Operating System (OS) level updates can be managed more easy an in an automated way. The default OS for these VM's is a Linux distribution with at least a container runtime installed on it (e.g. Docker, ContainerD, CRI-O). Although nowadays also Windows based VM's are supported, but this is currently out of scope.

Storage

Storage can be provided in several ways. A least some storage is attached to the VM's for the installation of the OS and memory swap space. For persistent storage of data used/generated by applications running on top of k8s, a Container Storage Interface (CSI) enabled storage needs to available. Although containers with in a k8s cluster can use the storage of the VM directly (host path), it is more preferable to use CSI storage via Persistent Volumes (PV) and Persistent Volume Claims (PVC). This prevents a container to be bound to a specific VM and enables k8s to be in control of scheduling containers to VM's.

Network - L2/4 Load Balancing

A k8s cluster needs to be reachable from within and outside a network, so applications running in the cluster can serve for example an API or a Web-app. A connection to one or more neighbouring networks needs to be established, so ip-addresses being served by the k8s cluster can be communicated to the neighbouring network's router.

DNS

Via the network and L2/L4 Load Balancing mechanisms only k8s cluster IP addresses are being reachable. It is more common to use human-readable domain names for applications being served in a k8s cluster. For this we need a Domain Name Server (DNS), preferably one that can integrate with k8s for automatically adding and deleting domain name records.

While the CBS wide DNS server (*cbsp.nl* domain) currently does not provide an API for automation, we have chosen to run a dedicated DNS server for k8s. This DNS server is being installed and managed by the Kubit team in a High-Available setup (Primary-Secondary) on two dedicated Linux VM's. We have chosen for the DNS Server application PowerDNS. This PowerDNS installation will act as an Authoritative DNS for the k8s subdomain/zone (e.g. *k8scloud.cbsp.nl*).

In case of a replacement of the CBS wide DNS server; additional requirements for k8s will be taken into account so the need for PowerDNS disappears.

Certificate Authority

Nowadays it is common practice to encrypt and secure connections, such as HTTPS. For this we need a Certificate Authority (CA) that is being trusted within (and maybe outside of) the network. This CA also generates and manages certificates for specific (sub-)domain names, so a specific URL can be served via HTTPS. Preferably the CA can integrate with k8s so certificates can be requested and managed automatically.