SNA Lab - HA

Beware choice 1 may not be the easiest

Choice 1 - HA tradition

Choose an HA cluster software and install two systems that are supported by it. For example if you choose HPE MC/SG[1], CentOS7 will do. Setup an identical service on both nodes, for example a simple HTML page saying on which node it lives (index.html:<p>sg1). Install the cluster software, configure the farm and validate that the simple service is redundant. *At least* brutally destroy a node, check the logs and see how the redundant application responds, live. No fencing nor shared storage are required.

Choice 2 - Load-balance intelligence

Consider a load-balancer, either layer 3 or HTTP reverse-proxy as mentioned during the lecture. Find out which one allows intelligent load-balance (by checking the status of its end-points). The end-points can be either different system nodes or just multiple service instances which are listening on different ports while keeping differenciating those e.g. with index.html:port8081. Do it and validate it (similarly as above).

Choice 3 - VRRP / CARP

Install two systems which are CARP-capable e.g. as NetBSD or GNU/Linux (ucarp) guests, then setup and validate IP redundancy with it. Show the VRRP/CARP packets flying around in wireshark or tcpdump. Crash a node (aka the *physical router*) and check that the IP and MAC is taken over by the redundant *virtual router*.

Choice 4 - pfsync

Try do make a Packet Filter (pf) router/gateway or firewall redundant by itself. The point here is to setup pfsync. Although it can be used in conjunction with CARP, we simply consider having multiple possible gateways but which act the same. For example when adding a port-forwarding rule, that should populate.

[1] HPE Serviceguard for Linux Trial Software https://www.hpe.com/emea_europe/en/resources/servers/serviceguard-linux-trial.html