

K3s-server-agent General Logic Flow

- A two node K3s cluster with etcd is deployed from Rancher
 - node-one is the server, node-two is the agent
- The cattle-cluster-agent on node-one creates a cronjob that periodically takes etcd snapshots and copies them to a specific location on the host filesystem of both nodes
- The cattle-cluster-agent on node-two periodically queries the kube-apiserver
 - If queries timeout sufficiently, the cattle-cluster-agent reports this to the Rancher Management Server
 - Rancher attempts to reach the K3s kube-apiserver
 - If successful, split-brain exists:
 - The kubelet on node-two is instructed to stop all pods
 - If unsuccessful, node-one is down:
 - Rancher deploys a plan that will start K3s on node-two in server mode using the most recent etcd snapshot
 - When node-one recovers, K3s will start in server mode
 - First step is to contact the cattle-cluster-agent on the peer node to check the current status
 - node-two is running as the server so node-one will start as an agent
- If node-two goes down, standard Kubernetes processes will recover