Distributed On-Demand Deployment for Transparent Access to 5G Edge Computing Services

UNIVERSITÄT KLAGENFURT

Josef.Hammer@aau.at, Hermann Hellwagner

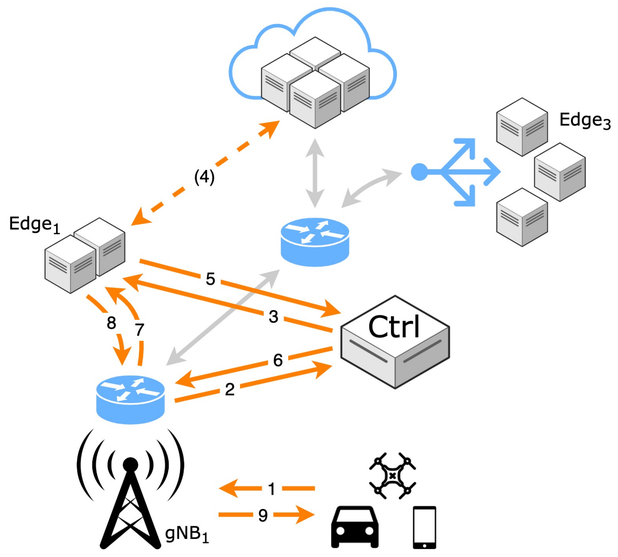
Context With *Transparent Access to Edge Services*, the request is redirected locally and ideally never reaches the cloud

Challenge The requested service might not be running yet at the local edge

Goal

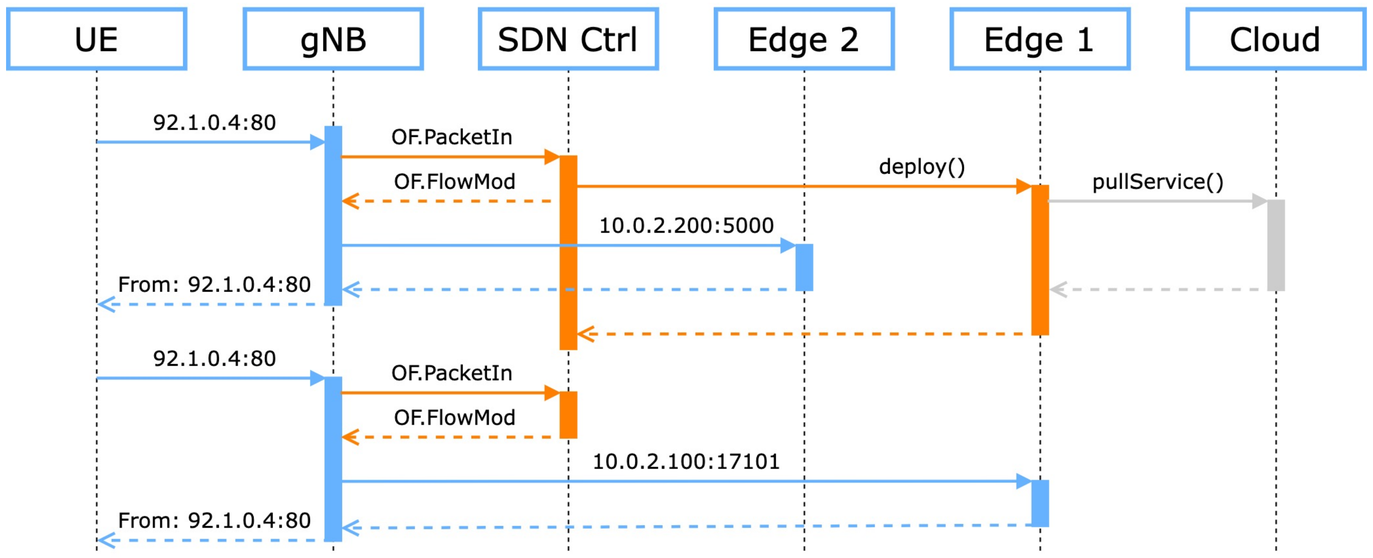
Is it feasible to keep a client's request to a non-running edge service on hold while deploying the *containerized service?*

With Waiting

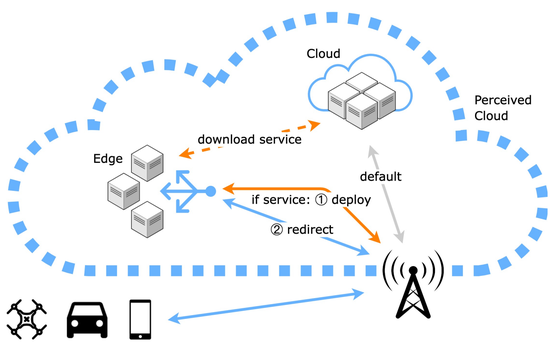


On-demand deployment with waiting: The user's request is kept waiting until a service instance has been deployed. The edge cluster might first have to pull the required service image from the cloud.

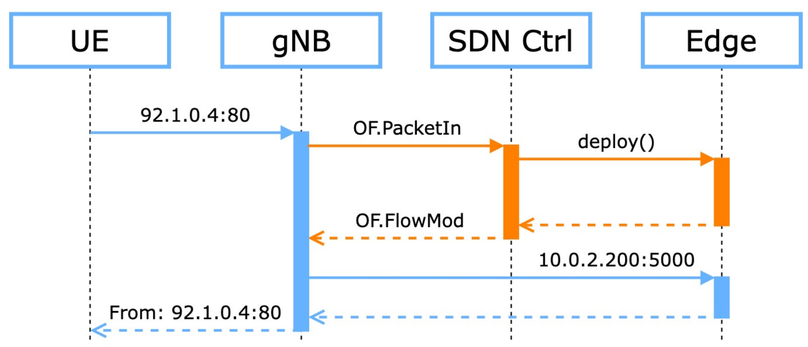
Without Waiting



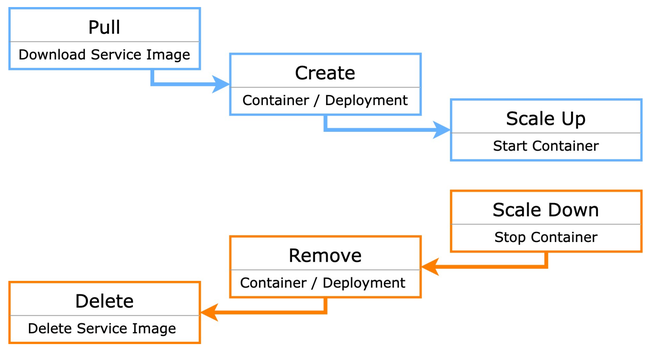
If the scheduler demands a very low response time, the SDN controller redirects the initial request to a running service instance in an edge further away. In parallel, the controller triggers the deployment of the service in the optimal edge. As soon as the new instance is running, requests are redirected to this optimal location.



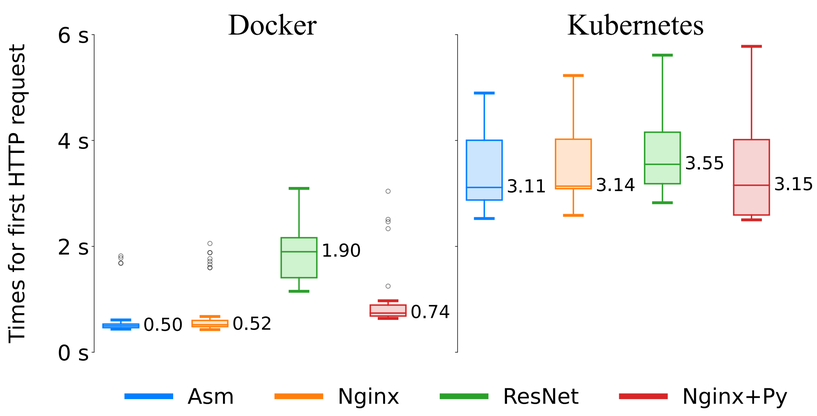
Transparent Access: All requests/responses look like cloud accesses to the client (UE) – the redirection to the edge is transparent



Routing with a registered service address (IP + port): Using OpenFlow (OF), the switch (gNB) transparently redirects the request to the edge server



Three deployment phases – for Kubernetes, we *create* a Deployment and Service with zero replicas and *scale up* separately

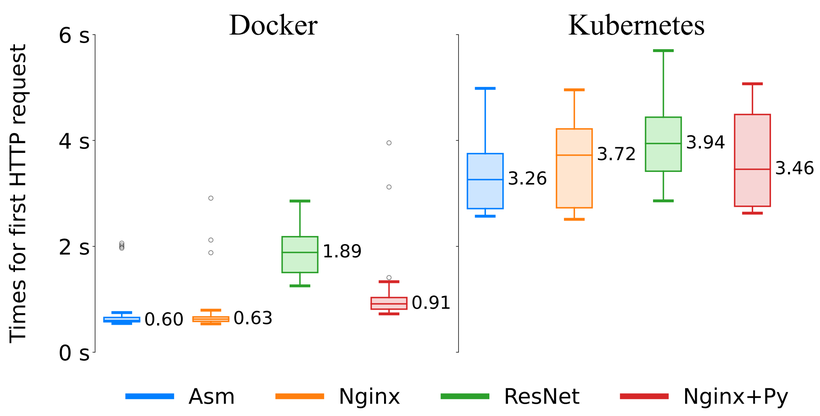


Experimental Results

Total time (median) to deploy four different services on two different clusters. We deployed 42 instances for each test. The numbers highlight the overhead of an orchestrator like Kubernetes (K8s).

Scale Up **Create + Scale Up**

Assembler Web Server – Nginx – TensorFlow Serving with ResNet50 – Nginx + Python App (2 containers)



J. Hammer and H. Hellwagner, "Distributed On-Demand Deployment for Transparent Access to 5G Edge Computing Services," 2023 IEEE Int. Parallel Distrib. Process. Symp. Work., 2023