October 08, 2019

Emulating in-cluster networking in an external workstation

Jimmy Cao

Emulating in-cluster networking in an external workstation

ABSTRACT

Machines external to a computer network typically access network resources via a proxy server internal to the network. However, external machines lack access to the network directory service; therefore, such machines cannot refer to internal network resources by their canonical names.

Per techniques of this disclosure, an external machine maintains a key-value file that acts as a directory service. When the external machine connects to the network, the key-value file is used to map network service names to the network address of the reverse proxy to resolve names of internal network resources. The reverse proxy uses the network resource name embedded in the service request of the external machine to forward the service request to the appropriate internal network resource. In this manner, a workstation external to a network emulates its presence within the network.

KEYWORDS

- Container orchestration
- Application container
- Cloud computing
- Remote access
- Application testing
- Production testing
- Domain name system (DNS)
- Proxy server
- Kubernetes

BACKGROUND

Machines (or services or computing resources) that are internal to a computer network can typically access resources within the network using a directory service such as DNS (domain name system). Internal machines or services can reference each other using canonical names that are mapped to network addresses through the directory service.

Machines that are external to a computer network typically access network resources by routing their requests via a proxy server internal to the network. The proxy server, e.g., a reverse proxy, lies within the network and has access to the directory service of the network. However, external machines lack access to the directory service; therefore, such machines cannot refer to internal network resources by their canonical names.

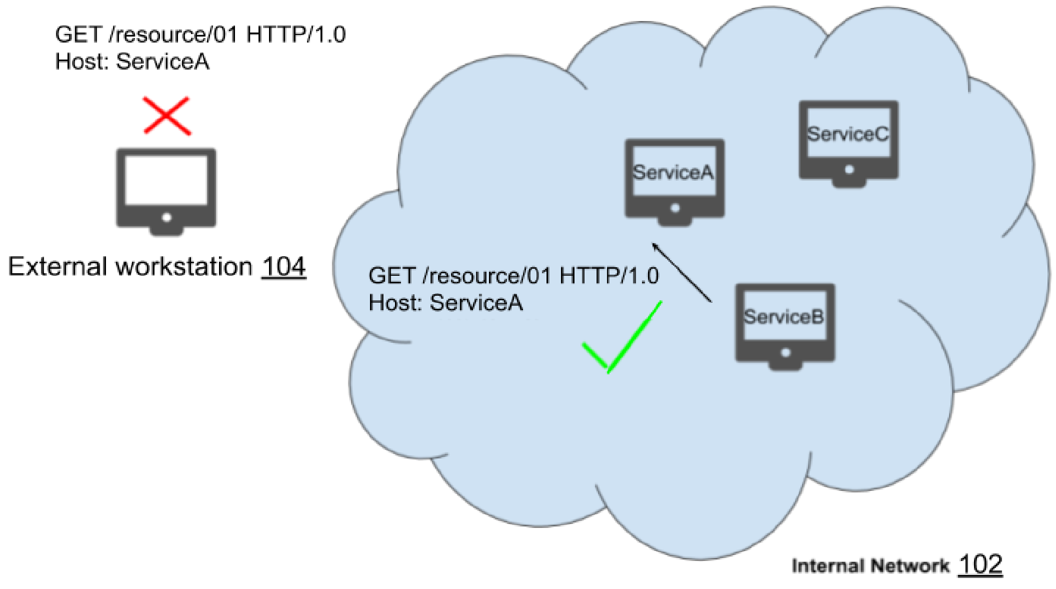

Fig. 1: An external workstation cannot use canonical names to access internal network resources

This is illustrated in Fig. 1, where computing resources (service A, service B, and service C) internal to a network (102) can access each other, for example by the `GET` command that service B uses to access service A. However, an external workstation (104) using the same `GET` command and including as parameter a named resource (service A) fails to access the resource.

DESCRIPTION

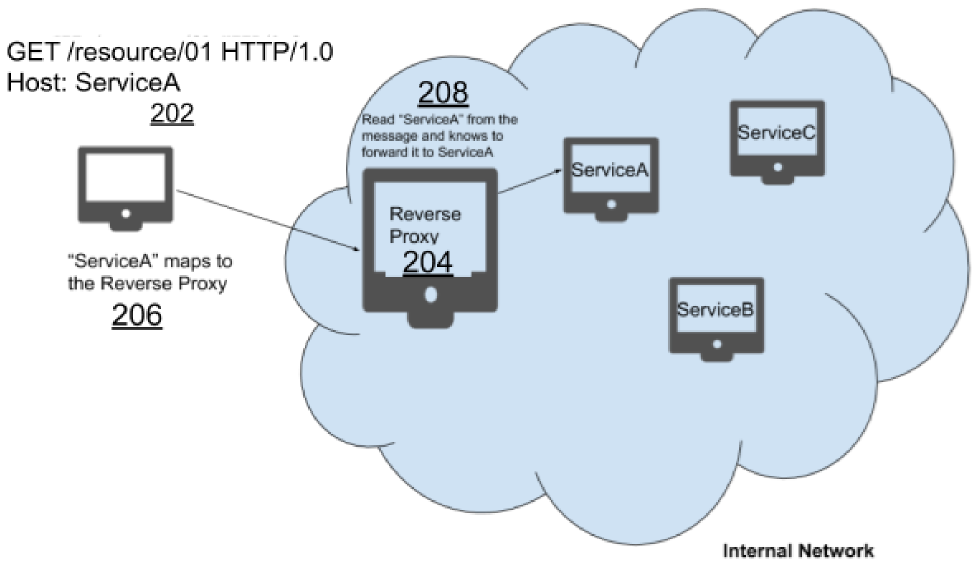
GET /resource/01 HTTP/1.0
Host: ServiceA
202
208
Read "ServiceA" from the message and knows to forward it to ServiceA
ServiceA
ServiceC
Reverse Proxy
204
"ServiceA" maps to the Reverse Proxy
206
ServiceB
Internal Network

Fig. 2: Enabling an external machine to access network resources using canonical names

Fig. 2 illustrates enabling external machines to access network resources using canonical names with the use of a reverse proxy. The external machine accesses the network by connecting to a reverse proxy server (204) located within the network. The external machine maintains a key-value file that serves as a directory service. Such a key-value file is similar to the `hosts` file of Unix. The keys of the key-value lookup table are kept synchronized with service names that exist within the directory service internal to the network. The key-value is used by the networking stack of the external machine to resolve canonical names. The key-value file maps internal service names, e.g., service A, service B, and service C in Fig. 2, to the network address of the reverse proxy (206).

The external machine accesses a resource within the network by embedding the name of the resource in the access request it sends to the proxy server, e.g., as illustrated in the `GET` command (202) of Fig. 1. The reverse proxy determines the intended recipient of the command by reading the name embedded within the access request (208). The reverse proxy forwards the request to the intended recipient, using the network directory service to resolve the name of the destination service. The reverse proxy also forwards the response of the destination service back to the external machine.

In this manner, an external machine can communicate with internal network services using their canonical names. The external machine thereby emulates its presence within the network.

The techniques herein facilitate the development of applications or services running within the internal network with quick setup and minimal configuration. For example, a software developer can run a service locally, e.g., on an external personal computer, and via the techniques described herein, connect it with live services running inside the network. A developer can test their application in a production environment, e.g., the ability of the application to communicate and integrate with internal network services, without having to manually deploy the application within the network. The techniques can be used, e.g., for application containers.

CONCLUSION

Per techniques of this disclosure, an external machine attempting to connect to a network maintains a key-value file of services or resources internal to the network. The external machine connects to a reverse proxy server that uses the network resource name embedded in the service request to forward the request to the appropriate internal network resource. In this manner, a workstation external to a network emulates its presence within the network.