

Kubernetes Networking Debugging with Netshoot

Using Netshoot to Debug Kubernetes Networking

Command:

```
kubectrl run tmp-shell --rm -i --tty --image=nicolaka/netshoot -- bash
```

Purpose:

This command launches a temporary interactive pod using the 'nicolaka/netshoot' image for advanced network debugging in Kubernetes.

Breakdown:

- `kubectrl run tmp-shell`: Create a temporary pod named tmp-shell.
- `--rm`: Automatically deletes the pod after you exit.
- `-i`: Keeps STDIN open for interaction.
- `--tty`: Allocates a terminal so you get a shell.
- `--image=nicolaka/netshoot`: Uses the 'netshoot' image, preinstalled with networking tools.
- `-- bash`: Opens a bash shell inside the container.

Tools Available in Netshoot:

- ping, traceroute, nslookup, dig, curl, tcpdump, ip, ss, ifconfig, nmap, iptables

Use Cases:

- Debug DNS issues inside pods
- Validate service-to-service communication via curl/ping
- Trace packet flow with tcpdump
- Identify NetworkPolicy or CNI misbehavior

Production Scenario Example:

You deployed 'orders-service' and your 'frontend' pod cannot reach it.

Steps:

1. Run:

```
kubectrl run tmp-shell --rm -it --image=nicolaka/netshoot -- bash
```

2. Inside pod:

```
curl http://orders-service.default.svc.cluster.local:8080/health
```

This verifies network path and service resolution.

Pro Tip:

Use aliases/scripts to quickly spawn netshoot pods in any namespace.