The ProxyJump pattern

You don't want to manually SSH to the bastion and then SSH again — that's clumsy and breaks tooling. SSH has a built-in feature for exactly this: **ProxyJump** (and its lower-level cousin, ProxyCommand). It transparently tunnels your connection to the internal host *through* the bastion, in one command:

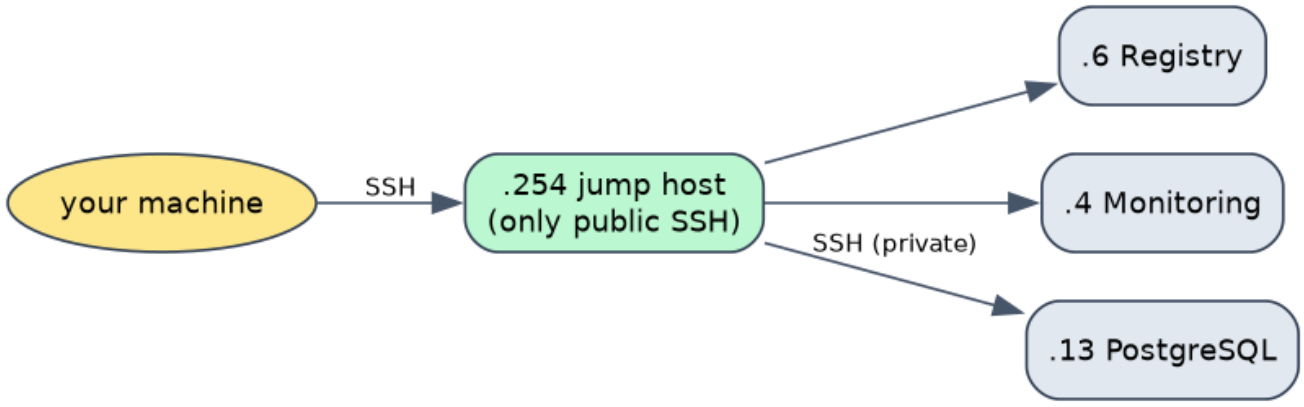
```
# conceptually:
ssh -J user@bastion  user@10.100.100.6      # land on the registry, via the jump host
```

To your terminal it feels like a direct connection; under the hood, the bastion is just forwarding the encrypted stream. The internal host's traffic never leaves the private network unencrypted, and you never have to think about the two-hop dance.

Everything administrative in this lab — every config change on an internal VM, every `kubectl` from a workstation, every database check — rode through the bastion this way. It's the connective tissue that makes a private-only network actually usable day to day.

> **Why we use this:** ProxyJump means "bastion-only access" costs you nothing in convenience. The security benefit (one door) usually comes with a usability tax; ProxyJump pays that tax for you. There's rarely a good reason to expose SSH on every box once you've got this in your fingers.

Diagram



Revision #2
Created 2026-04-19 13:03:02 UTC by Eslam Shapsough

Updated 2026-04-19 13:12:02 UTC by Eslam Shapsough