

ITI8801 · Cloud Architectures and DevOps

Recap, Teamwork & Networking Toolbox

Karl-Andreas Turvas, TalTech, Autumn 2026

Hi! 👋

If you haven't been to the practicals

Hi! 👋

I'm Karl-Andreas Turvas

DevOps Engineer at TalTech High Performance Computing Centre

Managing a cloud, a computing cluster...

...and teaching!

Have something intensive to calculate?

Tomorrow, 27.10.2026 from 13:30 - 16:30

HPC Centre Introduction for New Users event!

Registration deadline has passed already, but you may be able to squeeze in:



Or just write to us at hpcsupport@taltech.ee

So what am I teaching today?

1. Big picture recap
2. Transition to teamwork
3. Network design and debugging

Questions?

Ask right away!

or

Come to practicals to talk!

or

Shoot something my way over email or Teams: **katurv@taltech.ee**

Questions?

Ask right away!

or

Come to practicals to talk!

or

Shoot something my way over email or Teams: **katurv@taltech.ee**

~~"Oh I won't bother him..."~~ - Please do!

~~"Maybe it's not a smart question..."~~ - You're learning!

This is the second main value of this class for you!

Some housekeeping first!

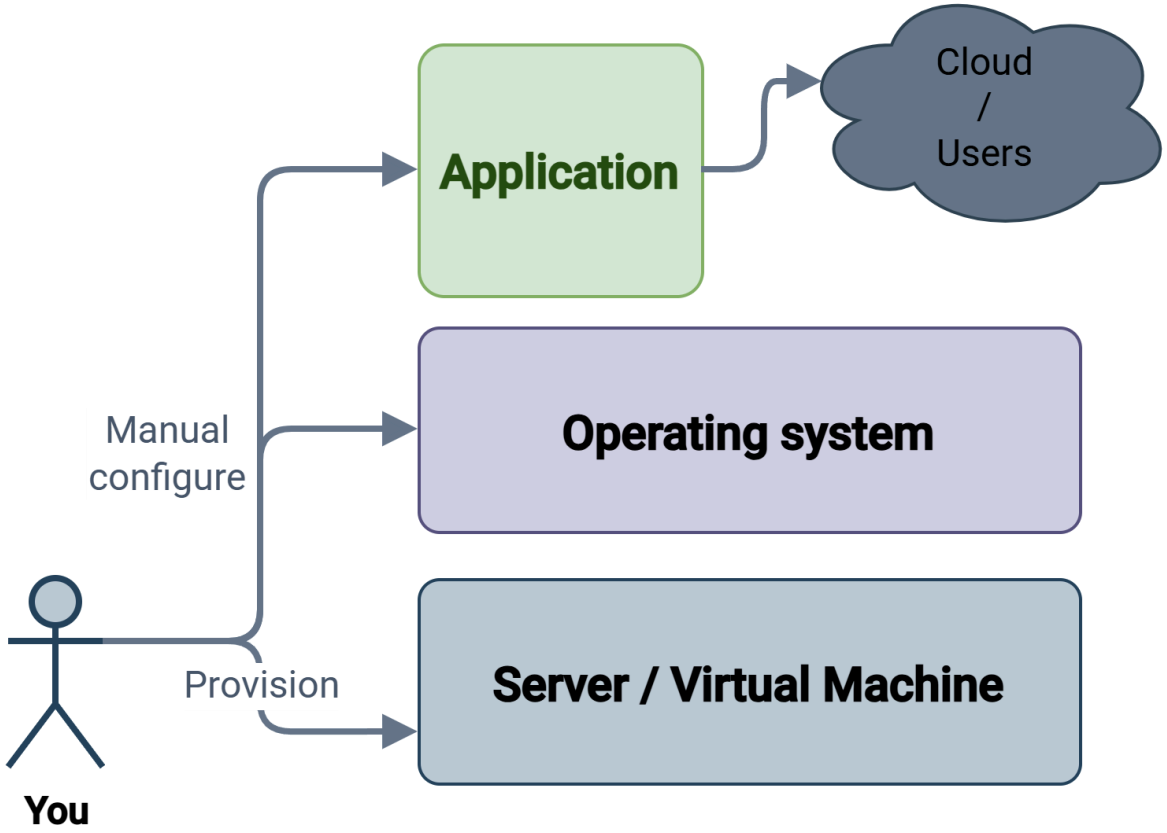
- University restricted a lot of infra (including part of this class' validator) to VPN/Estonian netblocks
- You may have gotten errors from the validator, but if you are confident about your work:
  - **Please still submit your homework and include a comment about the errors you encountered**
    - Timeouts, context deadline exceeded, etc point towards it being university issue
    - **Correctly done homework will be graded with full points even if you missed the deadline due to university networking issues**, but please mention the problem(s) when submitting the validator code
- It has been an interesting October 😅

Recap: Let's talk about you

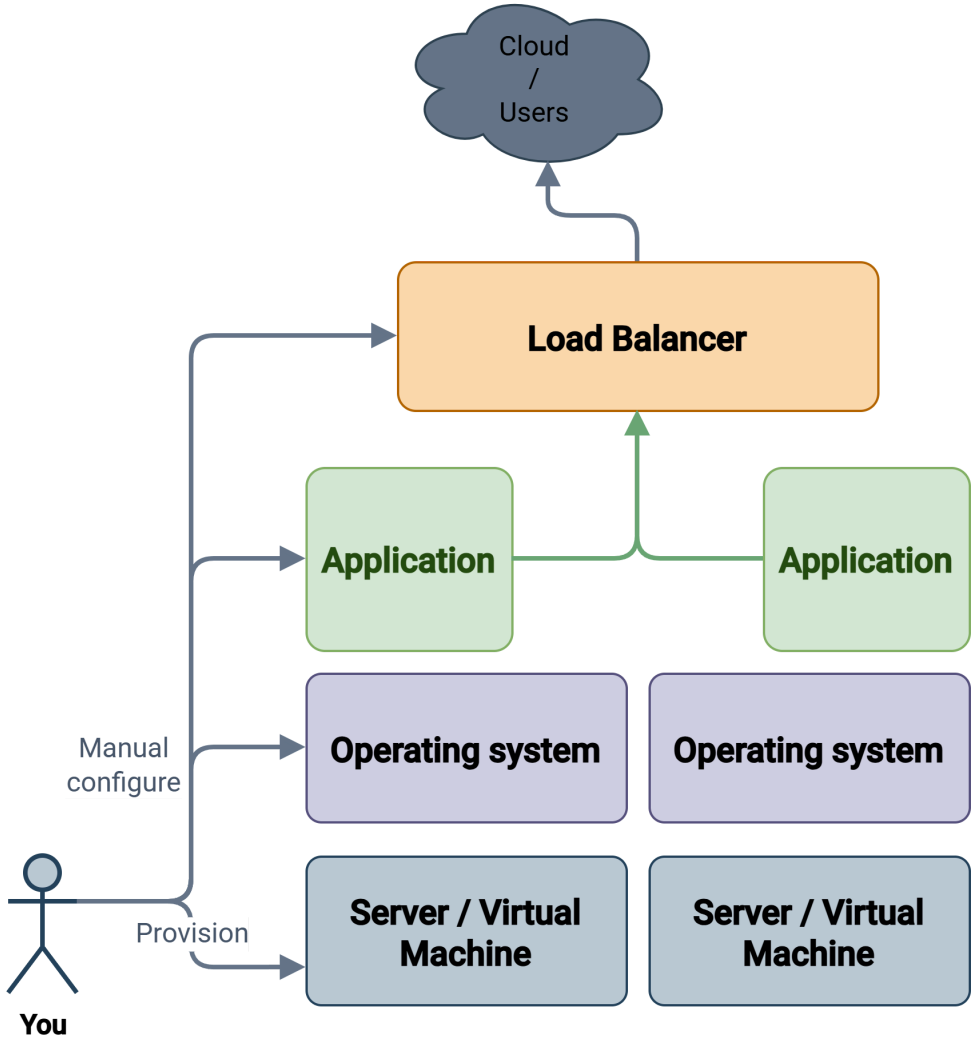


You

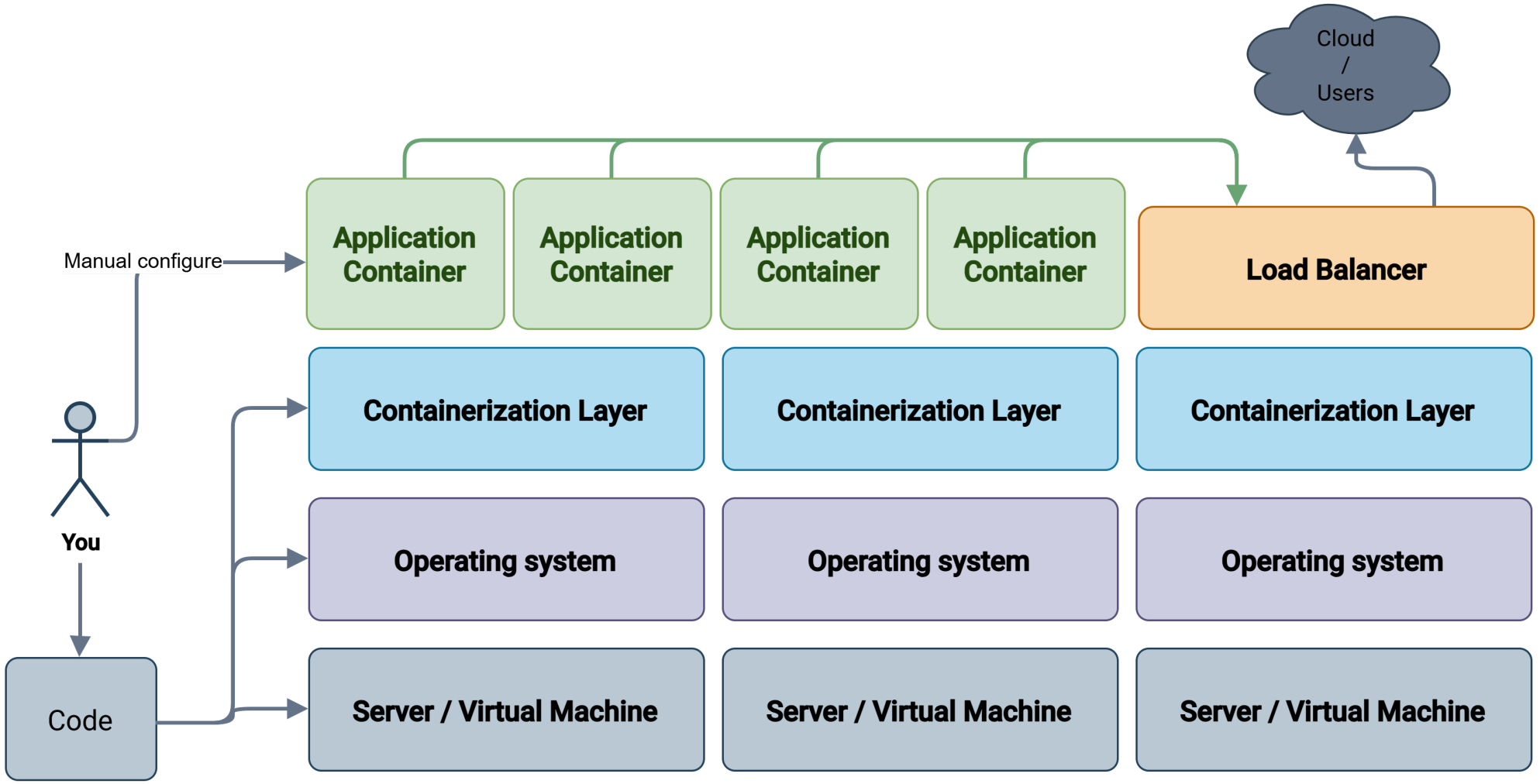
Recap: Lab 1 - Using Cloud



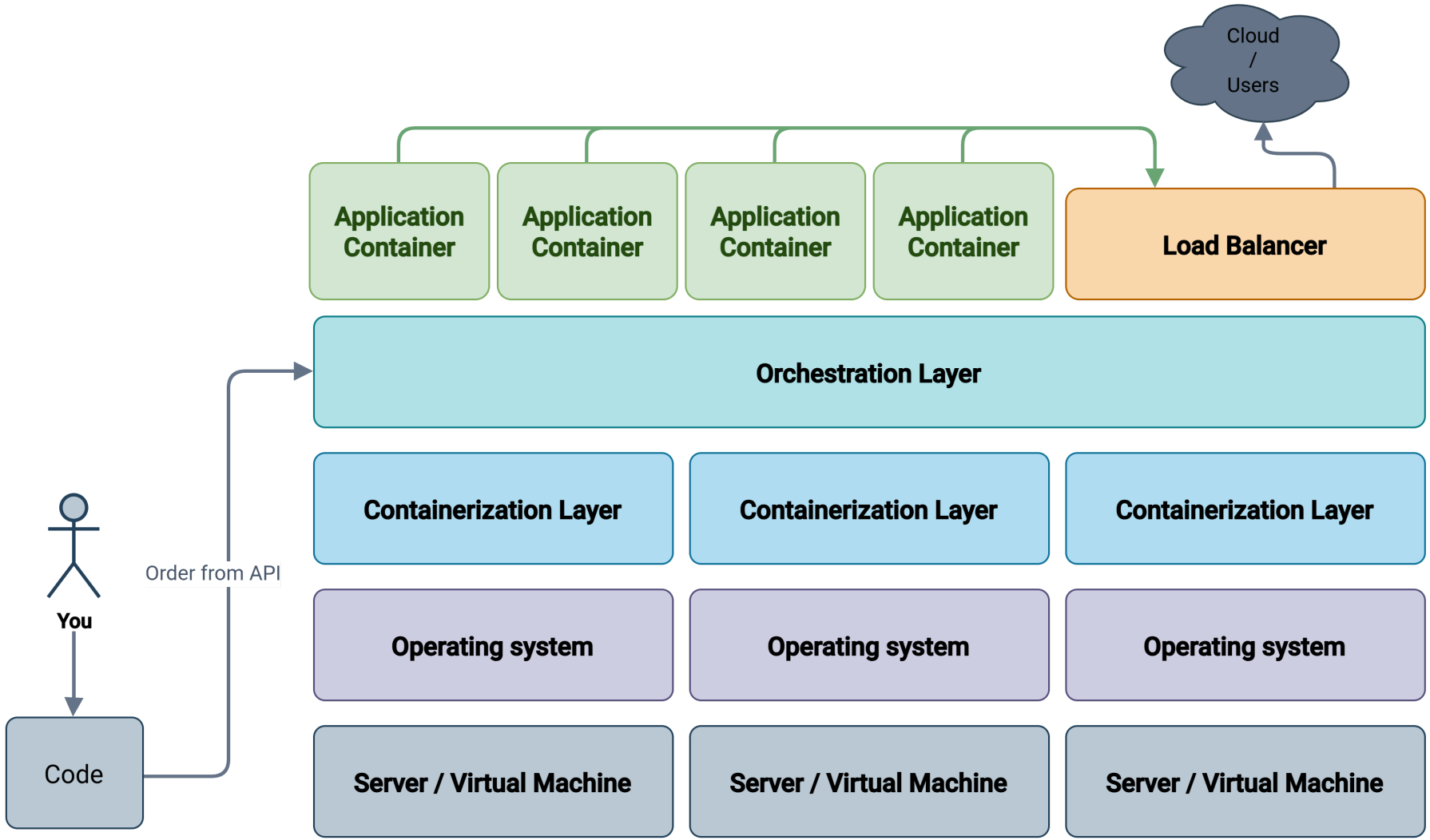
Recap: Lab 2 - Load Balancing



Recap: Lab 3-5 - Infra as Code



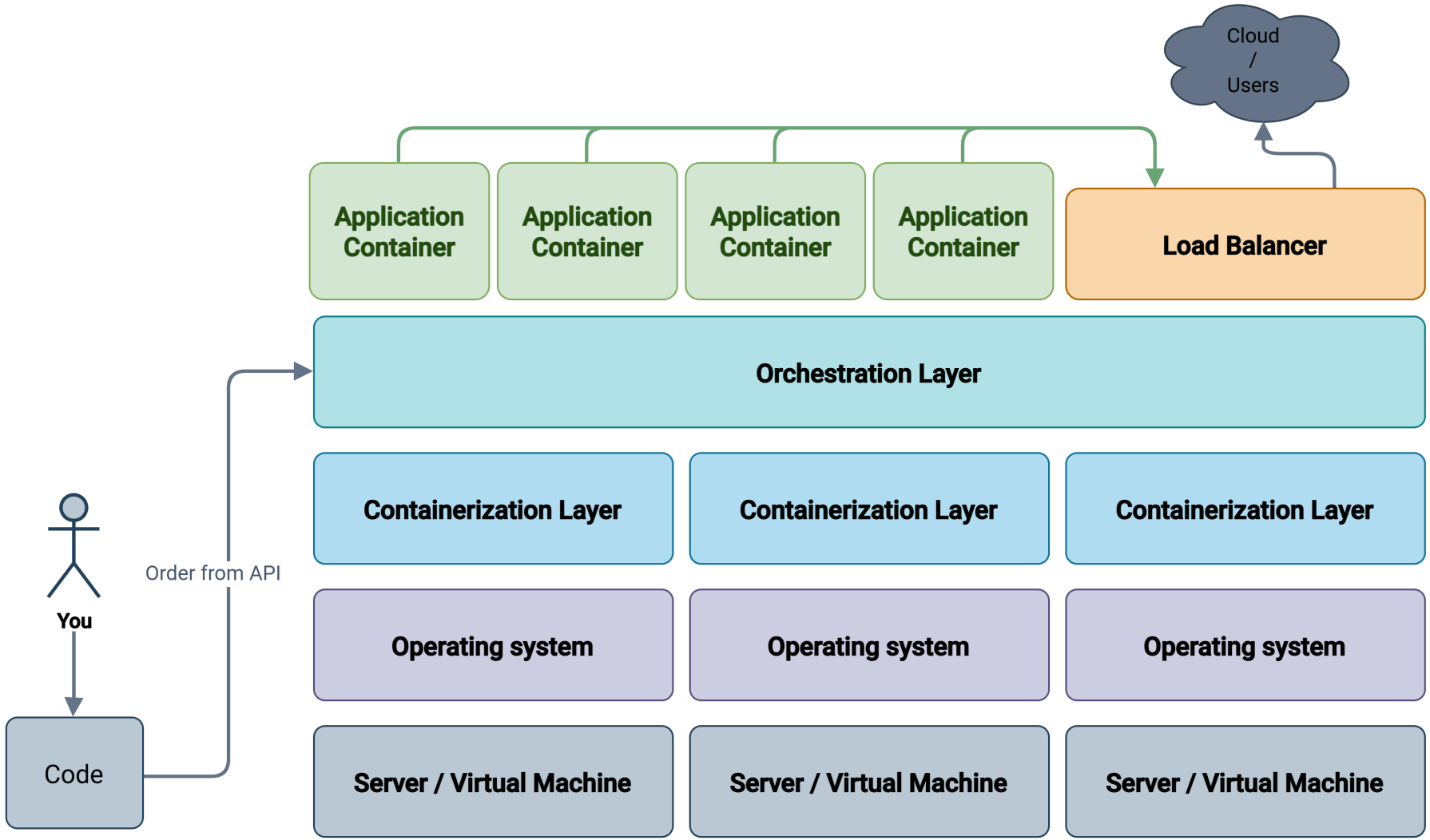
Recap: Lab 6-7 - Cloud Native Full Automation



Good job!

This stuff is not trivial!

But why this way?



But why this way?

To facilitate smooth teamwork at scale

1. Automation (imagine 10+ VMs, networks, firewalls, etc)
2. Documentation and transparency
3. Ergonomics, speed & security

Practice Sessions: "Teamwork at scale"

You will need to work as a team to set up a service

Goals:

- Illustrate the need for Infra as Code
- Verify what you have learned

See you at the lab for more details on this!

Now a bit about networking

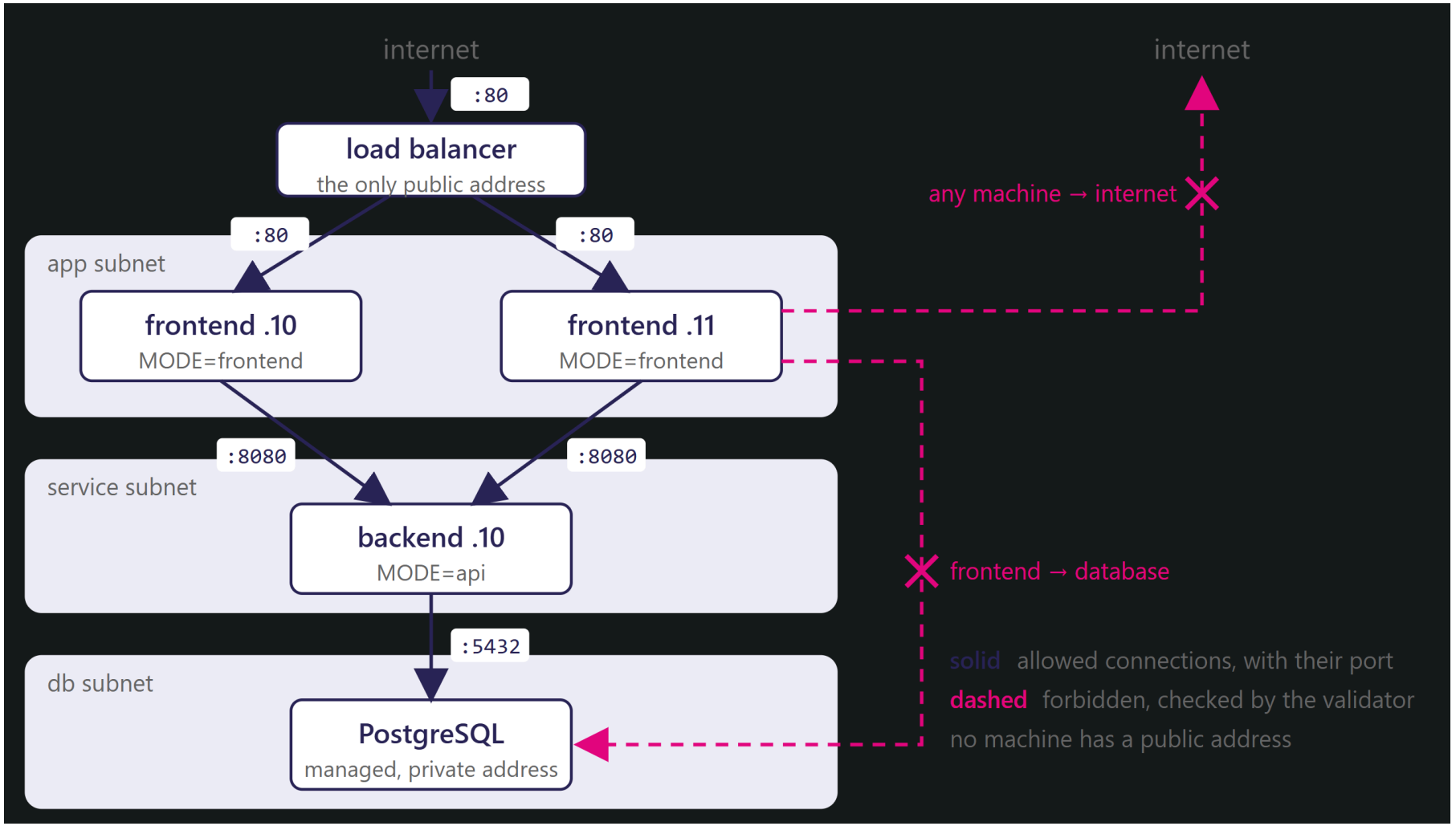
When you start thinking of the service and how to set up the networking:

- Review Lecture 2
- Think into the future
  - Risk analysis: most sensitive parts of the service
  - How to make it understandable for teammates
- Design! Think twice-thrice!
- Verify, verify, verify!

Unsure? **Ask us!**


LOOKS LIKE WE MADE A POOR DECISION.
NO. WE MADE A GOOD DECISION. THE OUTCOME WAS POOR.
workchronicles.com
THERE WERE MANY UNKNOWNS AND WE DECIDED BASED ON WHAT WE KNEW.
NEVER JUDGE A DECISION BY ITS OUTCOME.
Comics about work. Made with ♥ & lots of coffee.
Get the comics straight to your Inbox. Join the Newsletter.
Work Chronicles
workchronicles.com

A familiar picture, but ask why (again)



Security Design

Goals:

- **Confidentiality**: only authorized users can access sensitive information.
- **Integrity**: protect data from modification or corruption.
- **Availability**: services remain accessible when needed.
- **Manageability**: make the service easy to monitor, maintain, and update.

Apply that to all layers of the cloud

Think in layers:

1. **Physical infrastructure**: data centers, switches, routers, fiber.
2. **Virtual network**: isolated cloud networks and subnets.
3. **Connectivity**: routes, gateways, peering, VPNs, private links.
4. **Cloud Security**: firewalls, security groups, network ACLs, policies.
5. **Application**: load balancers, service discovery, DNS, service-to-service traffic.
6. **Operations**: logs, metrics, flow records, tests, and automation.

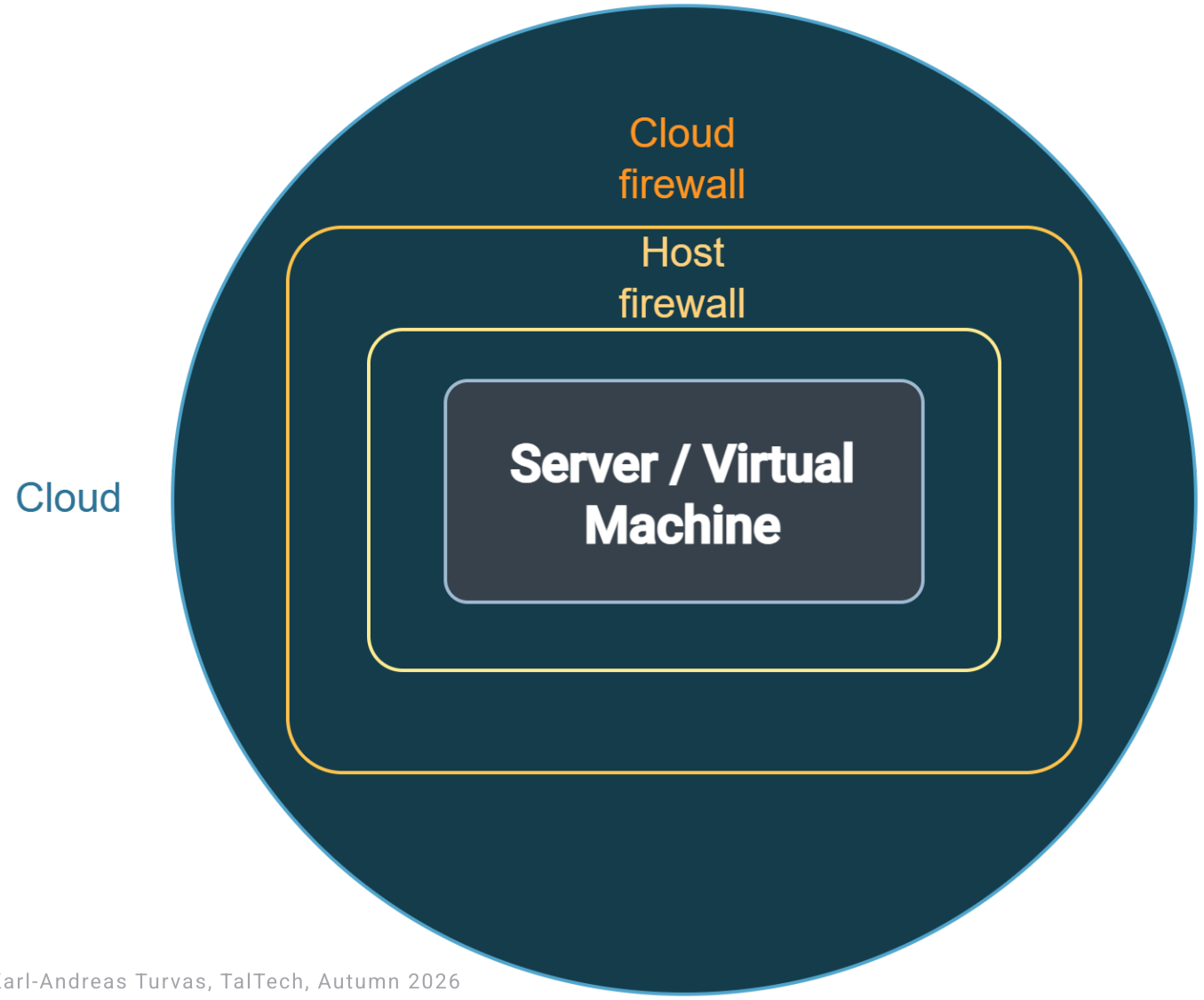
If you have gaps in networking knowledge: **Jim Kurose**'s **Computer Networking: A Top-Down Approach**

Common principles to follow

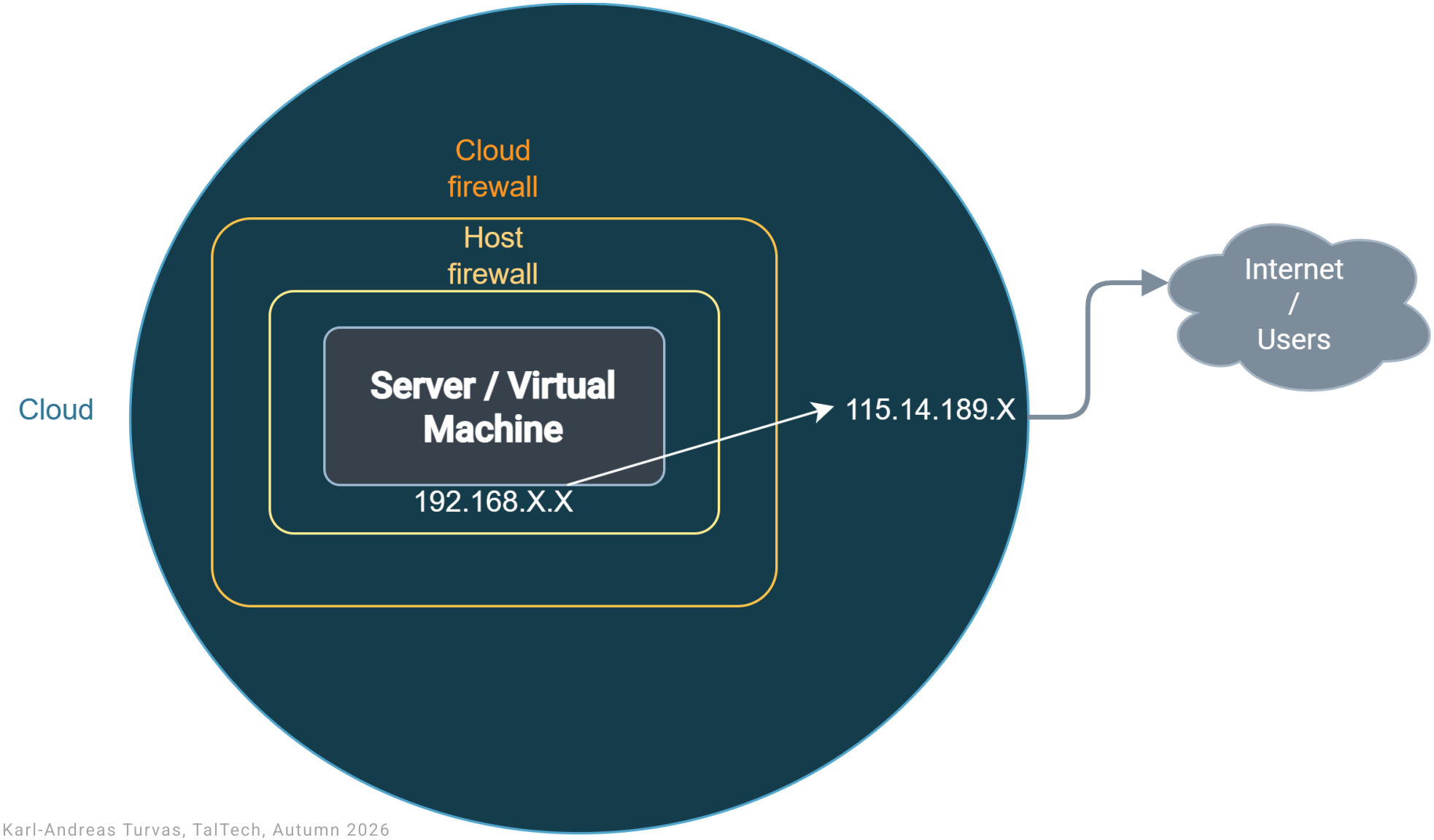
Design → Implement → **Double-Triple-Quadruple Check and Verify** → Repeat

1. Consider both cloud and host layer
2. Consider your perspective as admin and user's
3. Infrastructure as Code on all layers
4. Separate responsibilities = separate way in
    - Users, Devs, Ops, etc
5. Each tier (public, private, data) in their own subnet
6. Restrict bastion as much as possible
7. Minimize public endpoints

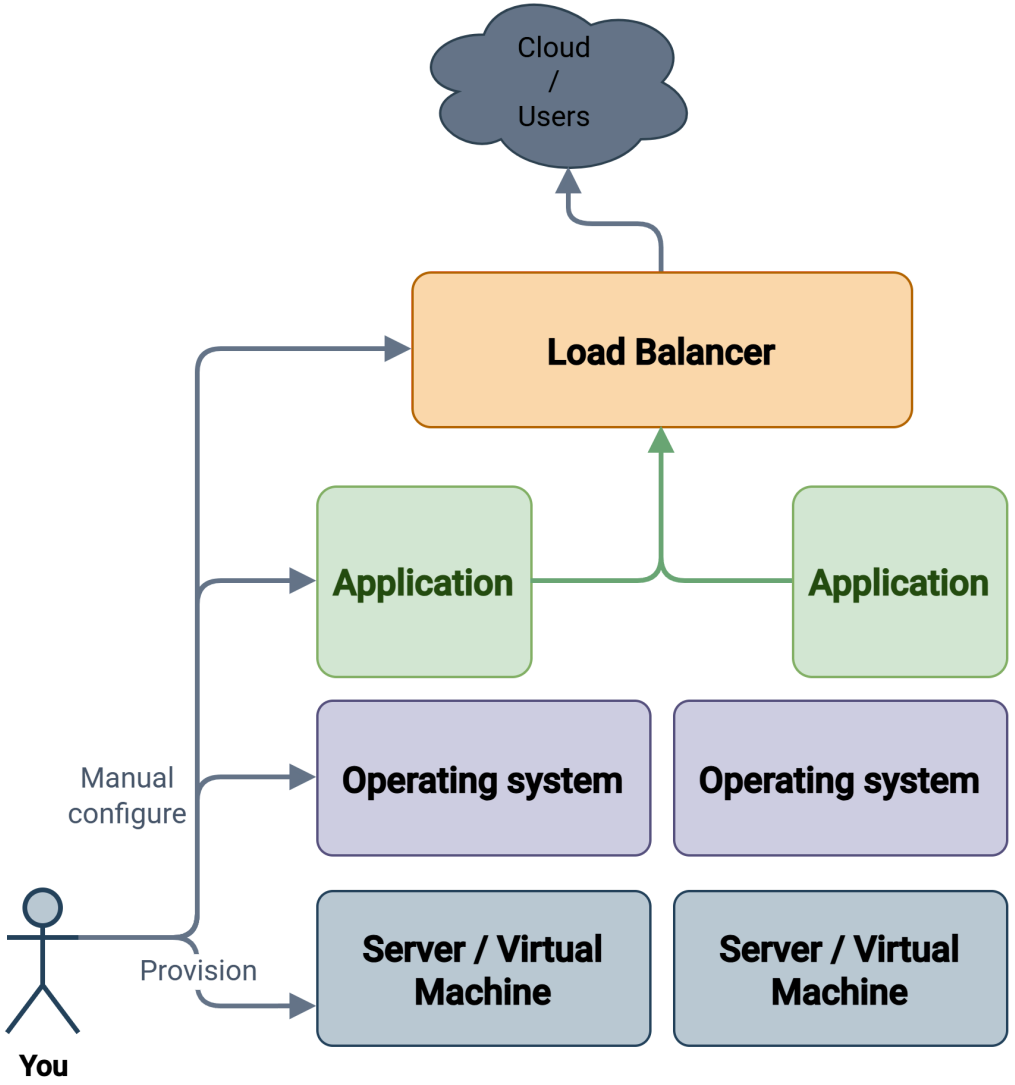
Cloud layering caveats: firewalls



Cloud layering caveats: firewalls and IPs



Way in and out should be separate



Way in (for you, the privileged admins)

Common approaches:

- **SSH**
  - Use and understand asymmetric keys!
- **VPN**
  - IP forwarding and firewall masquerading complexity

or indirect access:

- **Direct console** from cloud provider
  - How to do IaC? Pull mode?
- API based management, for example immutable OS, unikernels (Kairos, NixOS, etc)

- Bastion - disposable host used for administration

SSH: bot problems

- Not passwords, only keys!
- Restrict to known netblocks (VPN, TalTech, Estonia, etc)
- Fail2ban
- Credential management

This is something you have done in labs, but think it through!

VPN: Potential 2-layer problem again

If configuring Wireguard on a VM:

1. Which VM? Bastion?
2. New interface and a new subnet will be created for the VPN
    - Overlap?
3. IP forwarding may have to be enabled on host ***and*** cloud networking!
    - Forward from where to where?
4. MTU?
    - Bandwidth measurement with `iperf3` (Example: Wireguard Optimal MTU)
5. Credential management

The easier way: cloud providers *may* (wink, wink) have a way to directly order a "VPN"

General considerations with a bastion

Admin log in, monitoring, Ansible, etc

One way in, multiple ways out: connected to all subnets?

Very restricted IP/port filtering needed, like security group rules for bastion, etc

Debugging Toolbox

Remember:

> Design → Implement → ***Double-Triple-Quadruple*** **Check and Verify** → Repeat

Be careful of assumptions! Verify even if confident!

Debugging toolbox: IPs

Check IPs on both layers:

1. **Cloud local and public IPs**
2. Host interfaces and IPs: `ip address` (or `ip a` for short)
    - Remember Cloud routing, VIPs can be "invisible" from inside host!

Example output:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet 10.255.255.254/32 brd 10.255.255.254 scope global lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:15:5d:1e:dc:2b brd ff:ff:ff:ff:ff:ff
    inet 172.30.218.124/20 brd 172.30.223.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::215:5dff:fe1e:dc2b/64 scope link
       valid_lft forever preferred_lft forever
```

Debugging toolbox: Play Marco Polo!

Use `ping` to send ICMP protocol packets to a host

If it answers it will "pong" back

```
PING example.com (104.20.23.154) 56(84) bytes of data.
64 bytes from 104.20.23.154: icmp_seq=1 ttl=59 time=1.59 ms
64 bytes from 104.20.23.154: icmp_seq=2 ttl=59 time=1.55 ms
64 bytes from 104.20.23.154: icmp_seq=3 ttl=59 time=1.60 ms
64 bytes from 104.20.23.154: icmp_seq=4 ttl=59 time=1.36 ms
64 bytes from 104.20.23.154: icmp_seq=5 ttl=59 time=1.38 ms
^C
--- example.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3996ms
rtt min/avg/max/mdev = 1.358/1.494/1.596/0.102 ms
```

ICMP behaves differently to, for example, HTTP so `ping` might not be enough!

Debugging toolbox: Service

Is the name resolving and is connection encrypted?

Test TCP connectivity to HTTPS: `nc -vz example.com 443`

Inspect HTTPS behavior: `curl -v --connect-timeout 5 https://example.com/`

What is the difference between connecting via loopback, internal subnet IP, public load balancer IP?

Debugging toolbox: DNS

Check name resolve: `dig example.com`

```
; <<>> DiG 9.18.39-0ubuntu0.24.04.8-Ubuntu <<>> **example.com**
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52351
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;example.com.                   IN      A

;; ANSWER SECTION:
example.com.            213     IN      A       172.66.147.243
example.com.            213     IN      A       104.20.23.154

;; Query time: 9 msec
;; SERVER: 10.255.255.254#53(10.255.255.254) (UDP)
;; WHEN: Fri Oct 16 15:11:54 EEST 2026
;; MSG SIZE  rcvd: 72
```

Think: how will this differ from your own device "as user" and from local network?

Debugging toolbox: DNS

Is my active DNS server correct?

```
;; SERVER: 10.255.255.254#53(10.255.255.254) (UDP)
```

Check with another resolver: `dig example.com @<DNS SERVER>`

Does the result make sense?

```
;; ANSWER SECTION:
example.com.            213     IN      A       172.66.147.243
example.com.            213     IN      A       104.20.23.154
```

Debugging toolbox: DNS reverse lookups

Reverse lookup: `dig -x 172.66.147.243`

```
; <<>> DiG 9.18.39-0ubuntu0.24.04.8-Ubuntu <<>> -x 172.66.147.243
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;243.147.66.172.in-addr.arpa.   IN      PTR

;; AUTHORITY SECTION:
66.172.in-addr.arpa.    780     IN      SOA     cruz.ns.cloudflare.com. dns.cloudflare.com. 2366145569 10000 2400 604800 1800
```

Forward DNS and reverse DNS are two separate mappings!

In this case: website uses Cloudflare proxy so we don't see the origin IP

Debugging toolbox: DNS

Remember: DNS is designed to work in a chain!

Each resolver should forward the question until the one that knows sends back the answer.

So: what if you ask for internal IPs and you don't have DNS-Over-TLS configured?

Implement DNS in a weekend by Julia Evans

For fun: Rule 53 - If you can think of it, someone's done it in the DNS

Debugging toolbox: Routing

Check routing tables:

1. **Cloud routing tables!**
2. Host routing tables: `ip route` (or `ip r` for short)

```
default via 172.30.208.1 dev eth0 proto kernel
172.30.208.0/20 dev eth0 proto kernel scope link src 172.30.218.124
```

   - Traffic to `172.30.208.0/20` is sent **directly** through `eth0`.
   - Other IPv4 destinations use the default next hop `172.30.208.1`.
   - `src` indicates the preferred source address for that route.
3. Debug specific connections
   1. Only show routes from the given range of destinations `ip route get <SOURCE> to <DESTINATION>`
   2. Verify hops: `traceroute <DESTINATION>` (run multiple times)

Debugging toolbox: Tracing actual packets

Run `tcpdump` (like Wireshark!) on both hosts:

- Use filters:
  - Traffic only on specific host interface: `tcpdump -i <INTERFACE>`
  - Match one source, HTTPS destination: `tcpdump 'src 10.0.0.5 and dst port 443'`
  - etc
- Packets leave but don't arrive at the other end?
  - `SRC ↠ DST`
- Packets leave and arrive, but responses don't get back?
  - `SRC → DST ↠ SRC`

"I didn't need all this before!"

Maybe! Cloud providers may have convenient solutions to order:

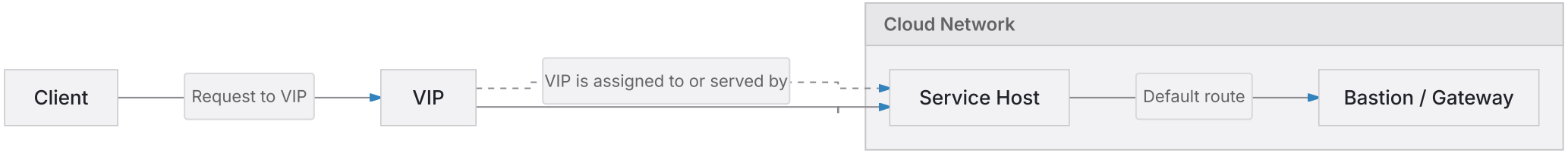
- Reverse proxies
- Bastions
- Load balancers
- VPNs

Foreshadowing 😉: What if your cloud provider does not have these options?

Funny example: Triangular Routing

Bastion was set up to provide SSH to admins and uplink as the default gateway to backend server.

Service used a public VIP to service users. And a separate VLAN with a bastion as gateway to service admins.



Different way in for admins and users: all good right?

But when I do `ping` and `curl` the VIP, no response: why?

Funny example: What to do?

On laptop:

1. Check external IP: `curl icanhazip.com` → `<LAPTOP_EXTERNAL_IP>`
2. Get route to service: `ip route get <VIP_IP>` →

   `<VIP_IP> via <LAPTOP_LOCAL_ROUTER_IP> dev eth0 src <LAPTOP_LOCAL_IP> uid 1000`
3. Set continuous ping: `ping <VIP_IP>`

On service host:

1. Listen for pings on VIP interface and filter only for laptop external IP:

   ```
   sudo tcpdump -ni <VIP_INTERFACE> -tttt \ 'icmp and src host <LAPTOP_EXTERNAL_IP>'
   ```

Funny example: Do we see the pings?

On laptop `ping` output:

```
PING <VIP_IP> (<VIP_IP>) 56(84) bytes of data.

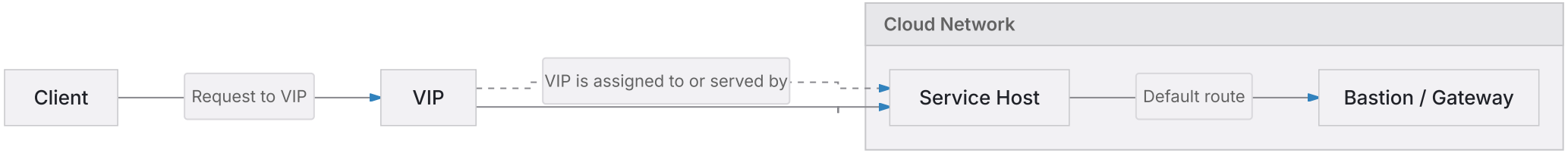
--- <VIP_IP> ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4098ms
```

On service host `tcpdump` output:

```
2026-10-19 10:15:01.100001 IP <LAPTOP_EXTERNAL_IP> → <VIP_IP>: ICMP echo request, id 4321, seq 1, length 64
2026-10-19 10:15:01.100250 IP <VIP_IP> → <LAPTOP_EXTERNAL_IP>: ICMP echo reply, id 4321, seq 1, length 64
2026-10-19 10:15:02.100102 IP <LAPTOP_EXTERNAL_IP> → <VIP_IP>: ICMP echo request, id 4321, seq 2, length 64
2026-10-19 10:15:02.100340 IP <VIP_IP> → <LAPTOP_EXTERNAL_IP>: ICMP echo reply, id 4321, seq 2, length 64
```

Funny example: Graphs are useful

Let's go back to the graph:



Let's check the route from the service host to laptop:

`ip route get <LAPTOP_EXTERNAL_IP>` →

`<LAPTOP_EXTERNAL_IP> via <BASTION_IP> dev eth0 src <SERVICE_HOST_IP> uid 0`

Funny example: So who responds?

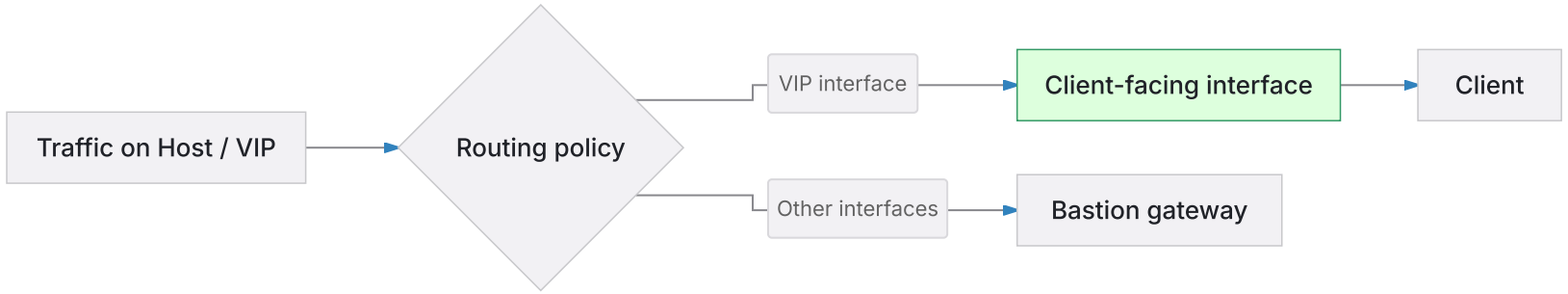
Which IP does the response come from?



Will the client even see a response that comes from a different IP?

Fix: Policy-Based Routing (PBR) aka "alternative routing tables"

Route VIP replies through the client-facing interface; keep the bastion as the default gateway for other traffic.



Create new `table 100`:

```
# Dedicated routing table for traffic sourced from the VIP
ip route add <CLIENT_SUBNET> dev <VIP_INTERFACE> table 100
ip route add default via <BASTION_IP> dev <VIP_INTERFACE> table 100

# Route packets sourced from the VIP using table 100
ip rule add from <VIP_IP>/32 table 100 priority 100

# Verify
ip route get <LAPTOP_EXTERNAL_IP> from <VIP_IP>
```

Overwhelmed? Draw!

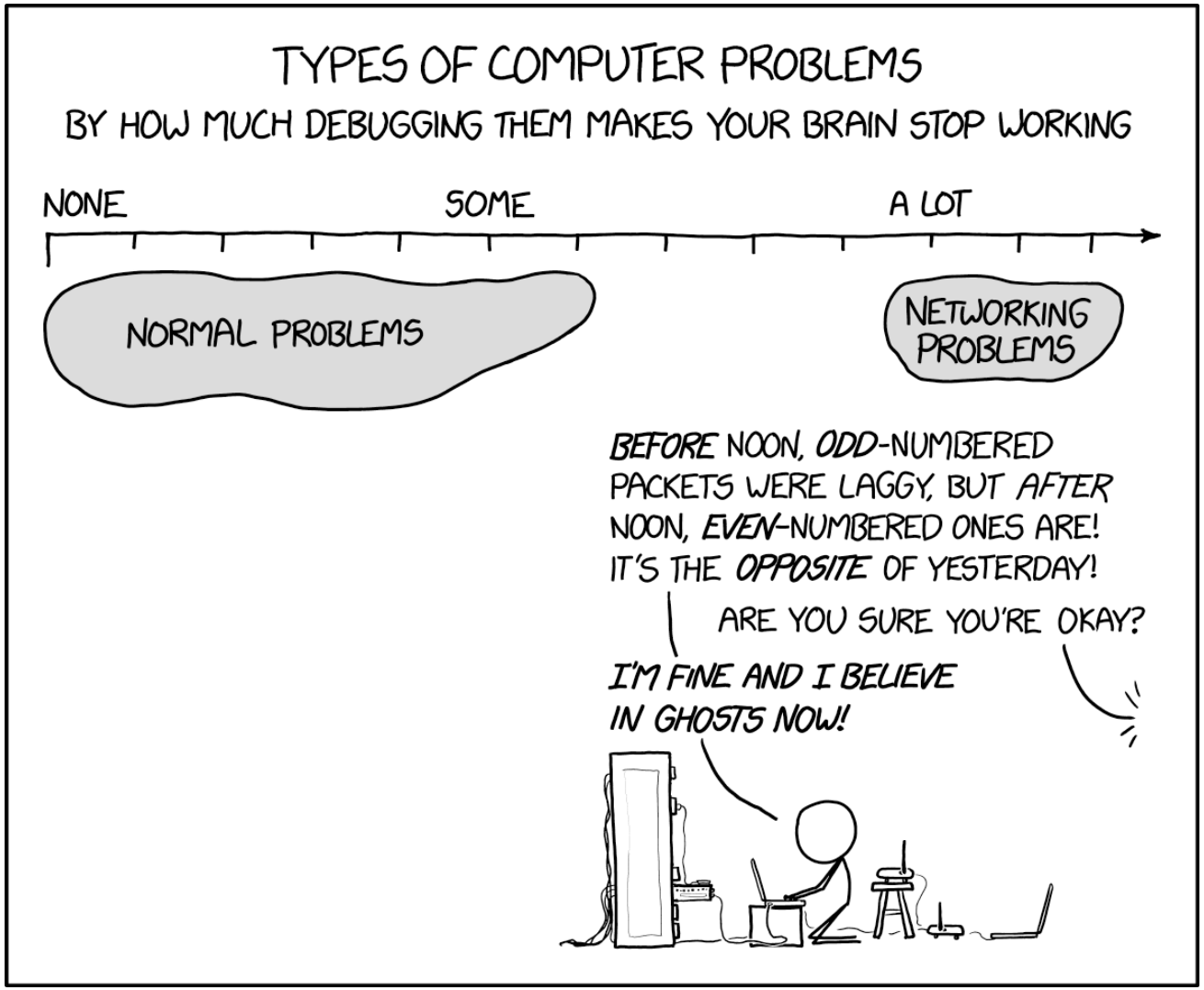
Graphs are useful!

Start with good old pen and paper...

...and then later digitize if needed:

- draw.io
- Mermaid graphs (markdown based)
- ASCIIFlow

Hopefully you will not need this toolbox a lot



Thank you for your time!

Any questions?

Contact me over email or Teams: **katurv@taltech.ee**

And come to practicals to talk/ask!