ICA0002: IT Infrastructure Services

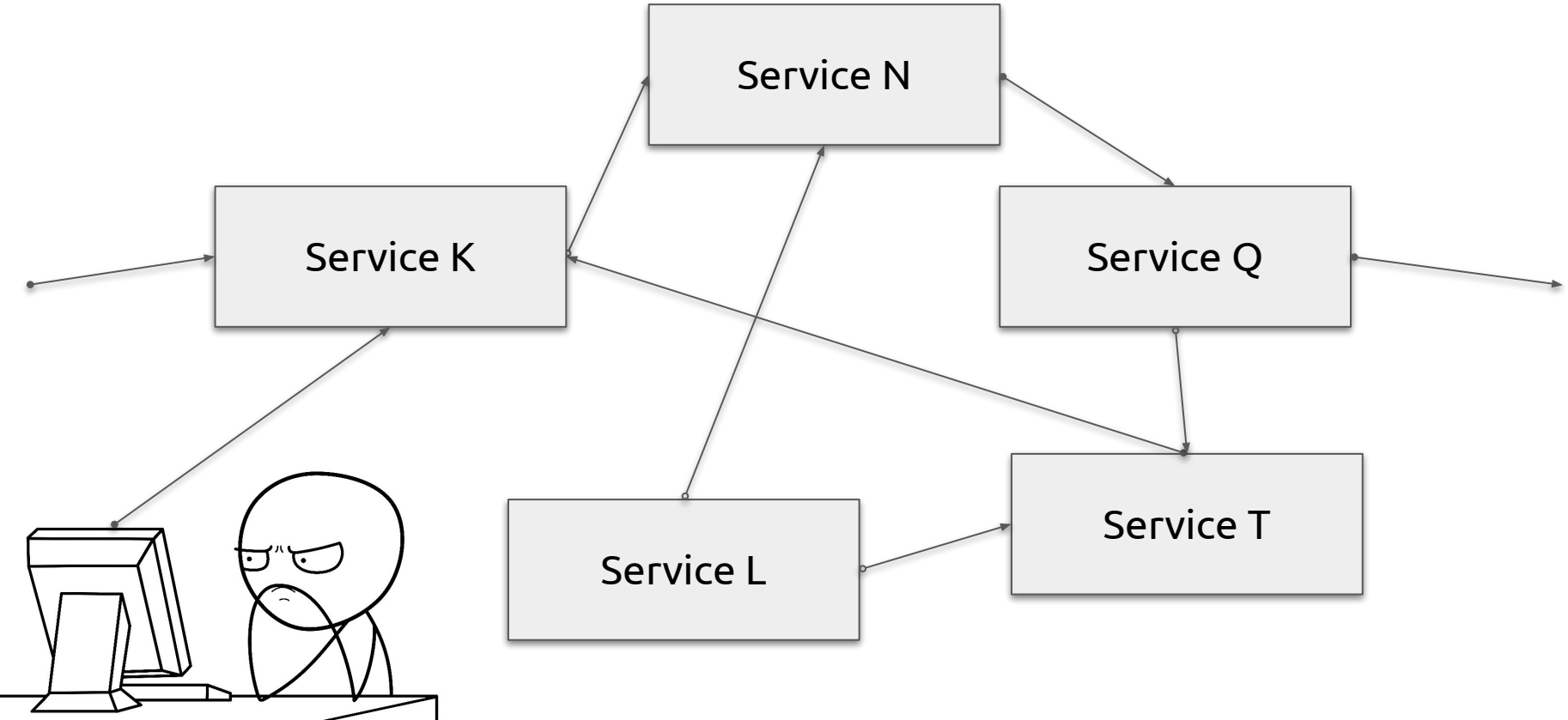
Troubleshooting

Roman Kuchin
Juri Hudolejev
2019

Troubleshooting Infrastructure Services

1. **Identify the misbehaving component**
2. Identify the problem
3. Find the acceptable solution
4. Fix it!

Identify the Misbehaving Component



Troubleshooting Infrastructure Services

1. ~~Identify the misbehaving component~~ ✓
2. **Identify the problem**
3. Find the acceptable solution
4. Fix it!

Identify the Problem

"I did everything correctly but it doesn't work!" -- is **not** a problem definition

Most common problems:

- Service is not running
- Service is running but not working correctly
- Service cannot communicate to another service

Problem:

Service is not running

Problem: Service is Not Running

How to detect, option 1:

```
ps ax | grep <my-service-name-part>
```

- **ps** - lists running OS processes
- **grep** - filter out all but your service

The simplest and fastest option to detect if the service is running.

Problem: Service is Not Running

Example command to check if process is running:

```
ps ax | grep apache
```

Example output if the process is running:

```
 883 ?        Ss   0:00 /usr/sbin/apache2 -k start
 902 ?        S    0:00 /usr/sbin/apache2 -k start
 903 ?        S    0:00 /usr/sbin/apache2 -k start
1636 pts/0    S+   0:00 grep --color=auto apache
```

Example output if the process is not running:

```
1638 pts/0    S+   0:00 grep --color=auto apache
```

Problem: Service is Not Running

How to detect, option 2:

```
systemctl status <my-service-name>
```

Provides more details about the service and also some logs.

Alternative if **`systemctl`** is not available:

```
service <my-service-name> status
```

Note: process, DEB package and Systemd unit names may differ!
Example: **`mysqld`** / **`mysql-server`** / **`mysql`**.

Problem: Service is Not Running

Example command to check service status:

```
systemctl status mysql
```

Example output if the service is running:

```
...
Active: active (running) since Sat 2019-10-26 11:51:03 UTC
...
```

Example output if the service is not running:

```
...
Active: inactive (dead) since Sat 2019-10-26 12:20:08 UTC
...
```

Problem: Service is Not Running

Most common causes:

- Service failed to start because of configuration issues
- Service failed to start because of lack of file permissions
- Service failed to start because the port in binds to is already taken
- Service did not start after the machine reboot
- You forgot to start the service :)

Configuration Syntax Problems

How to detect:

`apache2ctl -t` (or **`apache2ctl configtest`**)
`named-checkconf; named-checkzone`
`visudo -cf /etc/sudoers.d/my-user`

Service built-in config checkers.

The best way to check the configuration syntax of the existing files.

Only a few services have this option :(

Configuration Syntax Problems

Example command to check the service configuration syntax:

```
apache2ctl -t
```

Example output if the configuration syntax is correct:

```
...
Syntax OK
```

Example output if the configuration syntax is not correct:

```
AH00526: Syntax error on line 1 of /etc/apache2/sites-enabled/mysite.conf:
<VirtualHost> directive missing closing '>'
Action '-t' failed.
```

Configuration Syntax Problems

How to avoid:

```
template:
  src: mysite.apache2-conf.j2
  dest: /etc/apache2/sites-enabled/mysite.conf
  validate: apache2ctl -t
```

Validate the configuration **before** changing the actual file on server.

More examples for sudo, SSH etc.:

- https://docs.ansible.com/ansible/latest/modules/copy_module.html#examples
- https://docs.ansible.com/ansible/latest/modules/template_module.html#examples

Starting Service Automatically

Common practice in the Debian world:

- Start and enable the service automatically during DEB package installation

Do not rely on this behavior! This is just a convention, not a law.

The Ansible way to ensure that the service is started and enabled on boot:

```
service:
  name: apache2
  state: started
  enabled: yes
```

Problem:

Service is running but
not working correctly

Problem: Service Not Working Correctly

Examples:

- Key-based SSH is configured but it is still asking for a password on login
- Apache should listen on port 8080 but it is only listening on 80
- User **`root`** can log in to MySQL but user **`wordpress`** can not
- Bind should transfer entire zone but it only resolves individual addresses
- Script should backup these three directories but it backs up only this one
- etc.

Problem: Service Not Working Correctly

How to detect:

- No silver bullet exists -- every service has its own specifics
- Make sure that service **is actually running**
- Almost certainly a configuration **logic** problem; syntax is probably okay
- Localize the problem: it is easier is to find a solution for exact problems:
  - "Apache is not working" vs. "Apache is not listening on port 8080"
- Check logs
- Use verbose (debug) mode

Problem: Service Not Working Correctly

Logs:

- `/var/log/<service-name>`
- `/var/log/syslog`

Alternative commands for Systemd journal:

- `journalctl -fu <service-name>`
- `journalctl -xn`

Hints for Log Monitoring

Follow logs:

```
tail -f /var/log/apache2/error.log /var/log/syslog
```

Only print needed logs (containing **cron** in this example):

```
grep -i cron /var/log/syslog
```

Filter out some lines (print all but lines containing **systemd** in this example):

```
grep -v systemd /var/log/syslog
```

Problem: Service Not Working Correctly

Verbose (debug) mode:

- `ansible -v all -m ping`
- `curl -v http://localhost:8080`
- `wget -d http://localhost:8080`
- `ssh -vvv my-user@my-server`

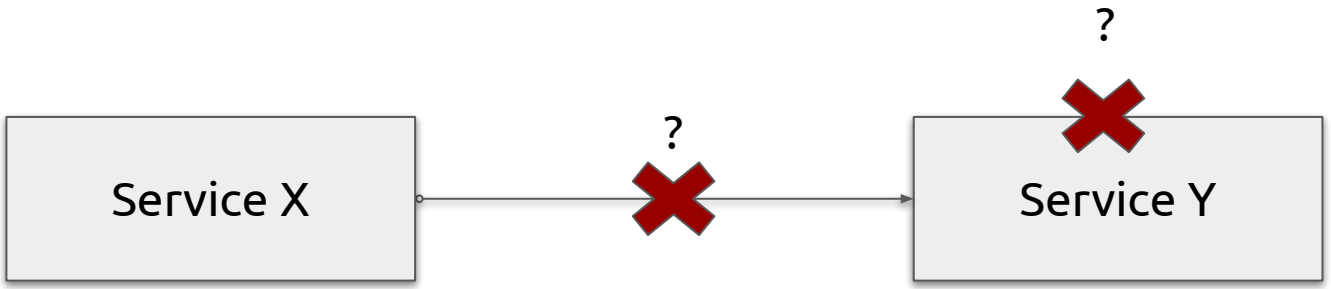
Apache: `LogLevel debug` in configuration file

^^^ you'll rarely need this

Problem:

Service cannot communicate to another service

Service Communication Issues



Service Communication Issues

First thing: make sure both services are **running** and **working correctly**.

Next thing: detect, identify and fix connectivity issues.

Common connectivity issues:

- Machine is not connected to the network
- Machine cannot reach another machine
- Service X cannot connect to another machine's port N
- Service Y is listening on wrong interface and/or port

Service Communication Issues

How to check if machine is connected to the network:

```
ping 1.1.1.1
```

Example output if the machine is connected (**Ctrl+C** to stop pinging):

```
64 bytes from 1.1.1.1: icmp_seq=3 ttl=56 time=9.70 ms
64 bytes from 1.1.1.1: icmp_seq=4 ttl=56 time=10.3 ms
…
```

Example output if the machine is not connected:

```
connect: Network is unreachable
```

Service Communication Issues

How to check if machine can reach another machine:

```
ping <another-machine-address>
```

Alternative if ping to remote machine is blocked by firewall:

```
traceroute <another-machine-address>
```

Service Communication Issues

How to check if one can connect to another machine's port:

```
nc -vz <another-machine-address> <port>
```

Alternative if **nc** is not available:

```
telnet <another-machine-address> <port>
```

Service Communication Issues

Example command to check if service is listening on the remote port:

```
nc -vz webserver 8080
```

Example output if the service is listening on the remote port:

```
Connection to webserver 8080 port [tcp/http-alt] succeeded!
```

Example output if the service is not listening on the remote port:

```
nc: connect to webserver port 8080 [tcp/http-alt] failed: Connection refused
```

Service Communication Issues

How to detect if service is listening on the correct port:

```
sudo netstat -lnptu
```

Needs admin privileges to list process names.

Alternative if **`netstat`** is not available:

```
sudo ss -lnptu
```

Service Communication Issues

Example command to list all services listening on TCP ports:

```
sudo netstat -lnpt
```

Example output if the service is listening on the remote port:

```
Proto  Recv-Q  Send-Q  Local Address    Foreign Address  State   PID/Program name
tcp    0       0       0.0.0.0:22       0.0.0.0:*        LISTEN  821/sshd
tcp    0       0       0.0.0.0:80       0.0.0.0:*        LISTEN  948/apache2
tcp    0       0       0.0.0.0:8080     0.0.0.0:*        LISTEN  948/apache2
```

Example output if the service is listening on the **local** interface only:

```
tcp    0       0       127.0.0.1:3306   0.0.0.0:*        LISTEN  946/mysqld
```

Troubleshooting Infrastructure Services

1. ~~Identify the misbehaving component~~ ✓
2. ~~Identify the problem~~ ✓
3. **Find the acceptable solution**
4. Fix it!

Finding the Solution

1. Check service documentation:
   - `<service-name> --help` example: `ansible --help`
   - `man <service-name>` example: `man apache2`
   - `info <service-name>` example: `info ssh`
2. Ask Google, DuckDuckGo, Bing etc.
3. Try rubber duck debugging -- not a joke, it really works
4. Ask a colleague for help

^^^ Strictly in **this** order!

Troubleshooting Infrastructure Services

1. ~~Identify the misbehaving component~~ ✓
2. ~~Identify the problem~~ ✓
3. ~~Find the acceptable solution~~ ✓
4. **Fix it!**

Questions?

Troubleshooting Cheat Sheet

Services and Processes

Process running?	`ps ax | grep <process>`
Service status:	`systemctl status <service>`
Config check:	`apache2ctl -t`
Verbose mode:	`ansible -v; ssh -vvv`
Service logs:	`/var/log/<service>/*` `journalctl -fu <service>`
System logs:	`/var/log/syslog` `journalctl -xn`
Help:	`<x> --help; man <x>; info <x>`

Connectivity

Network connected?	`ping 1.1.1.1`
IP address info:	`ip a`
Route info:	`route`
Reach other node:	`ping <ip>`
Probe remote port:	`nc -vz <ip> <port>`
Resolve hostname:	`host <hostname> [<ns>]`
Listening services:	`netstat -lnptu`