O RLY?

Stéphan Heisenbugs

Troubleshooting

Ok it's monday morning, you are super happy, you are at work early, and the first at work !
The slack channel for monitoring is full of messages saying servers are down, returning errors,...
Take a deep breath, let your heart calm down. Think twice before your intervention.

Keep calm

- Keep calm, you won't be fired
    - Even if you think you might be responsible, they should reflect on how you ended up messing up with their production.
- Don't just restart the server, find the root cause.
    - except if you know that something really wrong is going on and it's better to unplug everything : no service is sometime better than wrong/bad service.
- If after 30m/1h you don't have a clue than may be restarting can be a way of getting back to business.

> I know this doesn't sound webscale[tm], but you are not google.

Communicate & follow procedures

> Let's assume your company is not the perfect work environment, especially at hosting stuff. Some developers knows a bit but you don't really have a procedure in place to handle such incidents.

1. Communicate to users and manager, service desks :
    - setup dedicated slack channel #warroom
    - if you have status page, twitter account let's inform users that you have troubles and that your are investigating.
2. Apologize and commit to keep them updated
    - As a service provider, even if it's not your fault understand that you block their business
    - Keep them informed of the progress or non progress : nothing is more fustrating that no news.
3. Create a jira/github/... issue

4. Follow the procedure (if you have one)
    - Notify your manager,
    - assign an "incident lead" to handle the communication
    - find "experts" to help you test/confirm your analysis
5. Comment as you find information in the jira or slack channel
    - What is working
    - What is not working, when it last time reported working,…
    - What was modified lately (latest code modified, deployment, infra, …)
    - Paste commands/logs/screenshots that looks relevent
6. Try to find and document a workaround if applicable
    - eg : don't search by name, but by inventory code
7. Don't forget to update regularly status/twitter account/...

Reproduce - Diagnose - Fix - Reflect

Clarify and collect informations

- Where :
    - staging, production,...
    - client, server, db,...
    - which browser
    - which mobile device
- What :
    - What is the symptom ? Error, missing data, bad data, empty screen, never ending spinner, browser/app freezing,…
- How :
    - if only a part of the site is non functional, try document how to reproduce the issue
    - go to that page, fill in with this content, click search... boom
- Why :
    - if you have access to users, try understand the context, why the user was doing what he was doing when incident started.
    - Every beginning of the month, I'm supposed to send the report to accounting. I was trying to generate the monthly report but then everything went dark
- Evidences, collect all you can
    - Stacktrace, error message-codes
    - Thread dump, heap dump, core dump, tcp dump...
    - Logs (application, web server access logs)
    - Intermittent issue : it works but one request out of two is failing
    - Developer console errors in IE/Chrome/Firefox
    - Which queries are running on the database

> Automate evidence collections :
> - log forwarded to aggregator,
> - sentry alike service,
> - shell script to run jstack, collect db queries,
> - APM,
> ...

Try to locate the problem

- Use a dichotomic search : narrow the problem in 2 points of the stack, of the code.
- Use heuristic « gut feeling », perhaps based on latest incidents knowledge
- Use brute force in last resort

Try to relate to a previous issue or a change in your code or infrastructure

- a nightly unattended os update ?
- a new feature added to the product ?
- a new firewall rule added ?
- ...
- a new user behavior
- a user is abusing the system (github.com : commit message holding a dvd iso)

- marketing just sent user "Deals" or black friday ?

Make assumption and verify you have a safetynet, change one thing at a time

- make sure you have backups (db, config files,...)
- once you have mental model of the issue, test you hypothesis : with "readonly" commands
- avoid additional incidents, only one person change one thing at a time.
- if something needs to be changed, do one thing at a time to be sure on what was the corrective action.
- keep logs of commands and outputs for postmortem

Update the documentation / FAQ, Improve our tests

Document and reflect on the issue

- Do we have already an entry in our FAQ (or start one) ?
- Did you find something via google/stackoverflow ?
- Can we have similar problem in another service/app, in another team ?
- What can we improve to prevent this ?
- Do a post mortem on what to improve in your procedure

More on the subject

- http://www.catb.org/esr/faqs/smart-questions.html
- https://www.amazon.com/DevOps-Troubleshooting-Linux-Server-Practices/dp/0321832043
- https://pragprog.com/book/pbdp/debug-it
- https://jvns.ca/blog/2019/06/23/a-few-debugging-resources/
- https://twitter.com/b0rk/status/1146159551315677189
- https://twitter.com/b0rk/status/1145350304583622656
- https://twitter.com/b0rk/status/1146149101995778054
- https://twitter.com/b0rk/status/1148046227621199873
- https://twitter.com/b0rk/status/1144352412557283328
- https://twitter.com/b0rk/status/1148037549308481537
- https://twitter.com/b0rk/status/1147267035275104256
- https://twitter.com/b0rk/status/1147261917918109696
- https://twitter.com/b0rk/status/1146431464701124608

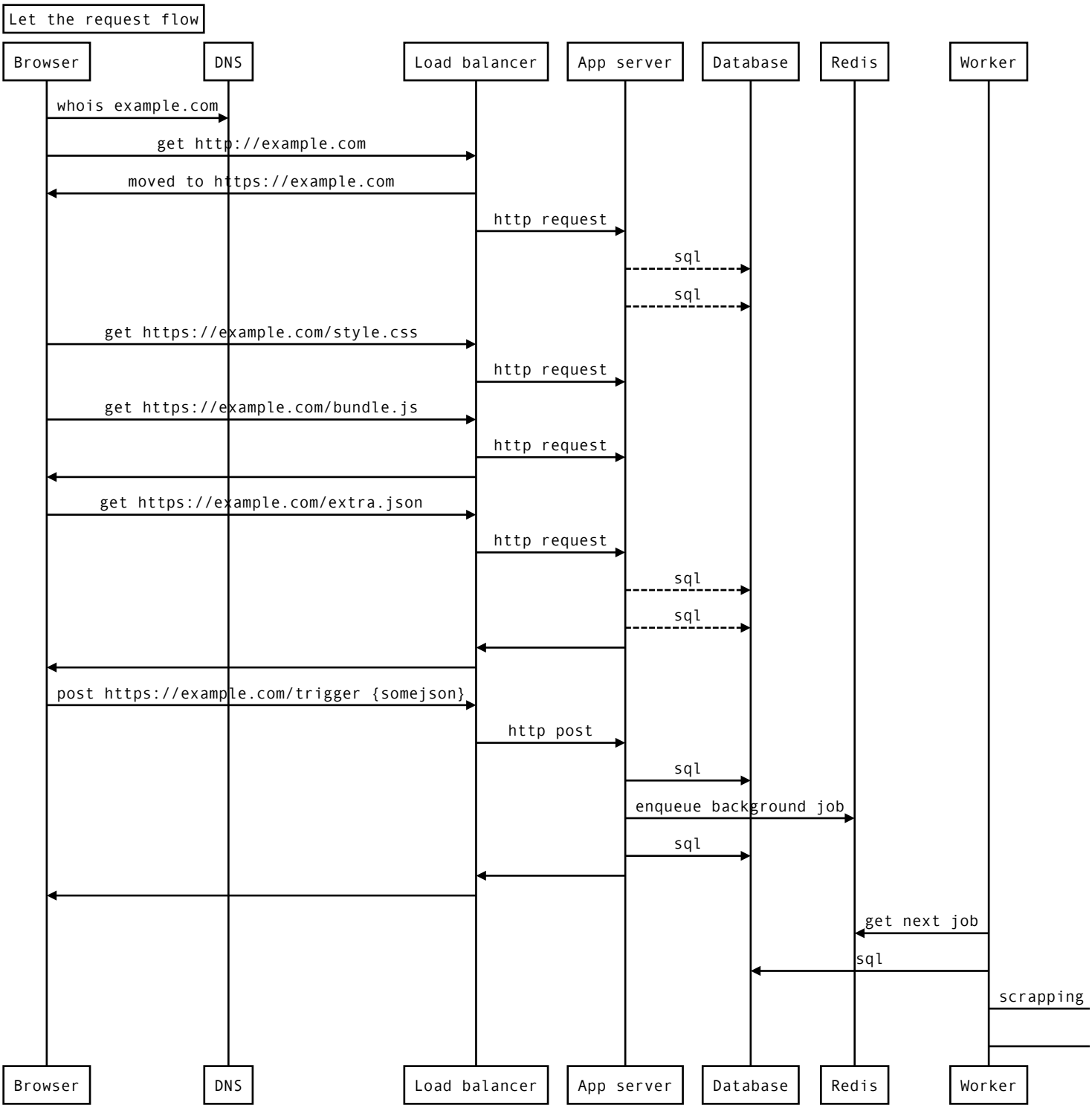
Introduction

What can go wrong ?

> Everything.
> -- said the developer

Let the music play

Accessing a website triggers plenty of activity on plenty of components/server



The Bad news

> Every arrow can fail, take hours to complete, never complete,...
> Every box can mis-behave, welcome to the internet.

The Good news

You are reading this book
the skills :

- awareness : most developers don't know the complete stack, never observe their application failing, learn from your mistakes
- evidence collection : get the logs, traces with the correct tool

WIP : Sample scenarios

- network
  - dns : no dns, ipv6, sometimes ok, sometimes not
  - tools
    - dig
    - pingdom (multisite)
- infra
  - diskspace : no space left on device
  - oom killer
  - ntpd ?
  - tools
    - df, du, iotop,...
- browser
  - ad blocker : /ad, /analytics,...
  - mixed content
  - csp
  - cors
- loadbalancer
  - bad gateway
  - http timeout
  - ddos, self-ddos
  - slow clients
  - heavy clients
  - queueing theory
- app
  - deadlock
    - lock
    - pool exhaustion (threads, db connections,...)
    - the importance of timeouts
  - slow
    - warm up
    - cache miss
    - verbose log level, synchronous, non buffered
    - regexp
  - infinite loop
    - cpu
  - garbage collection
    - pauses
    - plenty of object allocations
    - memory leaks
    - out of memory
  - strange bugs
    - fat client deployed from a shared drive
    - ftp active
  - zip/xml/... bomb
  - workers
    - paging
    - retries (becareful of too much retries)
  - tools :
    - thread dumps : kill -3
    - memory
    - application performance monitoring
- db
  - prepared statement
  - select n + 1
  - select * from db
  - orm vs sql

    - slow queries
    - non tuned memory/storage settings
    - tools :
        - explain plan
        - pgtop, mytop, db2top,... or their poor-man equivalent sql
- external services
    - degrade/fail gracefully

Networking

DNS what ?

In the first steps of network call a server need to translate a machine name to an IP address by asking a dns server if he knows about it. This might be the first step to fail too.

Known entry

Let's run a curl command to google.be and force ipv4, verbose logs

```
1  curl -vvv -4 google.be
2  * Rebuilt URL to: google.be/
3  *   Trying 216.58.211.99...
4  * Connected to google.be (216.58.211.99) port 80 (#0)
5  > GET / HTTP/1.1
6  > Host: google.be
7  > User-Agent: curl/7.47.0
8  > Accept: */*
9  >
10 < HTTP/1.1 301 Moved Permanently
11 < Location: http://www.google.be/
12 < Content-Type: text/html; charset=UTF-8
13 < Date: Tue, 09 Jul 2019 17:14:23 GMT
14 < Expires: Thu, 08 Aug 2019 17:14:23 GMT
15 < Cache-Control: public, max-age=2592000
16 < Server: gws
17 < Content-Length: 218
18 < X-XSS-Protection: 0
19 < X-Frame-Options: SAMEORIGIN
20 <
21 <HTML><HEAD><meta http-equiv="content-type" content="text/html;charset=utf-8">
22 <TITLE>301 Moved</TITLE></HEAD><BODY>
23 <H1>301 Moved</H1>
24 The document has moved
25 <A HREF="http://www.google.be/">here</A>.
26 </BODY></HTML>
27 * Connection #0 to host google.be left intact
```

At line `4`, you see that the name has been resolved to 216.58.211.99
At line `10`, the server tells you that `google.be` has been moved
At line `11`, the server tells you that you can follow `http://www.google.be/`

If you want to find how curl found the ip address, you can use dig

```
dig google.be

; <<>> DiG 9.10.3-P4-Ubuntu <<>> google.be
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1690
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;google.be.                     IN      A

;; ANSWER SECTION:
google.be.              255     IN      A       216.58.208.99

;; Query time: 31 msec
;; SERVER: 127.0.1.1#53(127.0.1.1)
;; WHEN: Mon Jul 08 21:05:14 CEST 2019
;; MSG SIZE  rcvd: 54
```

At line `15`, you find the address resolved by curl and it tells you that you can use that dns resolution for `255` seconds.

The challenge for dns server configuration is having a long enough time-to-live (ttl) to avoid extra dns lookup (and so slowing down the total request time) and short enough to allow fail over to take place (and redirect to another server in case of crash or overloaded network)

These changes might take times to be reflected in all dns servers/clients location, up to 48h if you specify a long ttl.

`whois` might find some extra infos about the owner and its dns server.

```
whois google.com
   Domain Name: GOOGLE.COM
   Registry Domain ID: 2138514_DOMAIN_COM-VRSN
   Registrar WHOIS Server: whois.markmonitor.com
   Registrar URL: http://www.markmonitor.com
   Updated Date: 2018-02-21T18:36:40Z
   Creation Date: 1997-09-15T04:00:00Z
   Registry Expiry Date: 2020-09-14T04:00:00Z
   Registrar: MarkMonitor Inc.
   Registrar IANA ID: 292
   Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
   Registrar Abuse Contact Phone: +1.2083895740
   Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
   Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
   Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
   Domain Status: serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited
   Domain Status: serverTransferProhibited https://icann.org/epp#serverTransferProhibited
   Domain Status: serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited
   Name Server: NS1.GOOGLE.COM
   Name Server: NS2.GOOGLE.COM
   Name Server: NS3.GOOGLE.COM
   Name Server: NS4.GOOGLE.COM
   DNSSEC: unsigned
   URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2019-07-09T14:48:20Z <<<
```

but note that whois has now less value since GDPR is here.

Unknown entry

If you try to curl a machine name that can't be resolved to an ip address

```
curl -vvv http://neverexisteddomains.com
* Rebuilt URL to: http://neverexisteddomains.com/
* Could not resolve host: neverexisteddomains.com
* Closing connection 0
curl: (6) Could not resolve host: neverexisteddomains.com
```

You might want to a bit more about that domain and use the dig command

```
dig neverexisteddomains.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> neverexisteddomains.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 43906
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;neverexisteddomains.com.       IN      A

;; Query time: 1 msec
;; SERVER: 127.0.1.1#53(127.0.1.1)
;; WHEN: Mon Jul 08 21:04:20 CEST 2019
;; MSG SIZE  rcvd: 41
```

The answer section is empty.

Who is might help find the owner, but when the entry is unknown nothing interesting will come up

```
whois neverexisteddomains.com
No match for "NEVEREXISTEDDOMAINS.COM".
>>> Last update of whois database: 2019-07-08T19:04:39Z <<<
```

and whois don't a records matching that domain.

Classic issues

Ends with .

Let's say you own the customdomain.be and add a CNAME record

```
domain IN CNAME myhosting.cloud.be
```

this will make `domain.customdomain.be` resolve to `myhosting.cloud.be.be.customdomain.be` which is probably not what you want.

You need to add an extra dot :

```
domain IN CNAME myhosting.cloud.be.
```

resolve `domain.customdomain.be` to `myhosting.cloud.be`

Propagation delay

https://www.quora.com/How-does-DNS-propagation-work

Aggressive cache

eg java will cache nearly forever, except if you tell him to not do so.

https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/java-dg-jvm-ttl.html

Some dns server do redirect

eg OVH offer "transparent" http redirect based on TXT dns records

this a mechanism to remember when debugging http request flow.

HTTPS

Self signed certificate

```
curl -vvv https://self-signed.badssl.com/
*   Trying 104.154.89.105...
* Connected to self-signed.badssl.com (104.154.89.105) port 443 (#0)
* found 148 certificates in /etc/ssl/certs/ca-certificates.crt
* found 597 certificates in /etc/ssl/certs
* ALPN, offering http/1.1
* SSL connection using TLS1.2 / ECDHE_RSA_AES_128_GCM_SHA256
* server certificate verification failed. CAfile: /etc/ssl/certs/ca-certificates.crt CRLfile: none
* Closing connection 0
curl: (60) server certificate verification failed. CAfile: /etc/ssl/certs/ca-certificates.crt CRLfile: none
More details here: http://curl.haxx.se/docs/sslcerts.html

curl performs SSL certificate verification by default, using a "bundle"
 of Certificate Authority (CA) public keys (CA certs). If the default
 bundle file isn't adequate, you can specify an alternate file
 using the --cacert option.
If this HTTPS server uses a certificate signed by a CA represented in
 the bundle, the certificate verification probably failed due to a
 problem with the certificate (it might be expired, or the name might
 not match the domain name in the URL).
If you'd like to turn off curl's verification of the certificate, use
 the -k (or --insecure) option.
```

```
curl -vvv -k https://self-signed.badssl.com/
*   Trying 104.154.89.105...
* Connected to self-signed.badssl.com (104.154.89.105) port 443 (#0)
* found 148 certificates in /etc/ssl/certs/ca-certificates.crt
* found 597 certificates in /etc/ssl/certs
* ALPN, offering http/1.1
* SSL connection using TLS1.2 / ECDHE_RSA_AES_128_GCM_SHA256
*        server certificate verification SKIPPED
*        server certificate status verification SKIPPED
*        common name: *.badssl.com (matched)
*        server certificate expiration date OK
*        server certificate activation date OK
*        certificate public key: RSA
*        certificate version: #3
*        subject: C=US,ST=California,L=San Francisco,O=BadSSL,CN=*.badssl.com
*        start date: Wed, 12 Jun 2019 15:31:59 GMT
*        expire date: Fri, 11 Jun 2021 15:31:59 GMT
*        issuer: C=US,ST=California,L=San Francisco,O=BadSSL,CN=*.badssl.com
*        compression: NULL
* ALPN, server accepted to use http/1.1
> GET / HTTP/1.1
> Host: self-signed.badssl.com
> User-Agent: curl/7.47.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: nginx/1.10.3 (Ubuntu)
< Date: Mon, 08 Jul 2019 19:03:07 GMT
< Content-Type: text/html
< Content-Length: 477
< Last-Modified: Fri, 21 Jun 2019 23:18:34 GMT
< Connection: keep-alive
< ETag: "5d0d65ca-1dd"
< Cache-Control: no-store
< Accept-Ranges: bytes
<
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="shortcut icon" href="/icons/favicon-red.ico"/>
  <link rel="apple-touch-icon" href="/icons/icon-red.png"/>
  <title>self-signed.badssl.com</title>
  <link rel="stylesheet" href="/style.css">
  <style>body { background: red; }</style>
</head>
<body>
<div id="content">
  <h1 style="font-size: 12vw;">
    self-signed.<br>badssl.com
  </h1>
</div>

</body>
</html>
* Connection #0 to host self-signed.badssl.com left intact
```

External assessment

https://www.ssllabs.com/ssltest/

Test your client library

https://badssl.com/

ssh

Permission denied

```
ssh bad@github.com
Permission denied (publickey).
```

Which certificate is used to authenticate ?

```
ssh -vvv bad@github.com
OpenSSH_6.6.1, OpenSSL 1.0.1k-fips 8 Jan 2015
debug1: Reading configuration data /etc/ssh/ssh_config
debug1: /etc/ssh/ssh_config line 56: Applying options for *
debug2: ssh_connect: needpriv 0
debug1: Connecting to github.com [140.82.118.3] port 22.
debug1: Connection established.
debug1: identity file /home/ec2-user/.ssh/id_rsa type -1
debug1: identity file /home/ec2-user/.ssh/id_rsa-cert type -1
debug1: identity file /home/ec2-user/.ssh/id_dsa type -1
debug1: identity file /home/ec2-user/.ssh/id_dsa-cert type -1
debug1: identity file /home/ec2-user/.ssh/id_ecdsa type -1
debug1: identity file /home/ec2-user/.ssh/id_ecdsa-cert type -1
debug1: identity file /home/ec2-user/.ssh/id_ed25519 type -1
debug1: identity file /home/ec2-user/.ssh/id_ed25519-cert type -1
debug1: Enabling compatibility mode for protocol 2.0
debug1: Local version string SSH-2.0-OpenSSH_6.6.1
debug1: Remote protocol version 2.0, remote software version babeld-93408c70
debug1: no match: babeld-93408c70
debug2: fd 3 setting O_NONBLOCK
debug3: load_hostkeys: loading entries for host "github.com" from file "/home/ec2-user/.ssh/known_hosts"
debug3: load_hostkeys: found key type RSA in file /home/ec2-user/.ssh/known_hosts:1
debug3: load_hostkeys: loaded 1 keys
debug3: order_hostkeyalgs: prefer hostkeyalgs: ssh-rsa-cert-v01@openssh.com,ssh-rsa-cert-v00@openssh.com,ssh-rsa
debug1: SSH2_MSG_KEXINIT sent
debug1: SSH2_MSG_KEXINIT received
debug2: kex_parse_kexinit: curve25519-sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-
debug2: kex_parse_kexinit: ssh-rsa-cert-v01@openssh.com,ssh-rsa-cert-v00@openssh.com,ssh-rsa,ecdsa-sha2-nistp256-cert-v01@openss
debug2: kex_parse_kexinit: aes128-ctr,aes192-ctr,aes256-ctr,arcfour256,arcfour128,aes128-gcm@openssh.com,aes256-gcm@openssh.com,
debug2: kex_parse_kexinit: aes128-ctr,aes192-ctr,aes256-ctr,arcfour256,arcfour128,aes128-gcm@openssh.com,aes256-gcm@openssh.com,
debug2: kex_parse_kexinit: hmac-md5-etm@openssh.com,hmac-sha1-etm@openssh.com,umac-64-etm@openssh.com,umac-128-etm@openssh.com,h
debug2: kex_parse_kexinit: hmac-md5-etm@openssh.com,hmac-sha1-etm@openssh.com,umac-64-etm@openssh.com,umac-128-etm@openssh.com,h
debug2: kex_parse_kexinit: none,zlib@openssh.com,zlib
debug2: kex_parse_kexinit: none,zlib@openssh.com,zlib
debug2: kex_parse_kexinit:
debug2: kex_parse_kexinit:
debug2: kex_parse_kexinit: first_kex_follows 0
debug2: kex_parse_kexinit: reserved 0
debug2: kex_parse_kexinit: ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-sha256
debug2: kex_parse_kexinit: ssh-dss,rsa-sha2-512,rsa-sha2-256,ssh-rsa
debug2: kex_parse_kexinit: chacha20-poly1305@openssh.com,aes256-gcm@openssh.com,aes128-gcm@openssh.com,aes256-ctr,aes192-ctr,aes
debug2: kex_parse_kexinit: chacha20-poly1305@openssh.com,aes256-gcm@openssh.com,aes128-gcm@openssh.com,aes256-ctr,aes192-ctr,aes
debug2: kex_parse_kexinit: hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha2-256,h
debug2: kex_parse_kexinit: hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha2-256,h
debug2: kex_parse_kexinit: none,zlib,zlib@openssh.com
debug2: kex_parse_kexinit: none,zlib,zlib@openssh.com
debug2: kex_parse_kexinit:
debug2: kex_parse_kexinit:
debug2: kex_parse_kexinit: first_kex_follows 0
debug2: kex_parse_kexinit: reserved 0
debug2: mac_setup: setup hmac-sha1-etm@openssh.com
debug1: kex: server->client aes128-ctr hmac-sha1-etm@openssh.com none
debug2: mac_setup: setup hmac-sha1-etm@openssh.com
debug1: kex: client->server aes128-ctr hmac-sha1-etm@openssh.com none
debug1: kex: ecdh-sha2-nistp256 need=20 dh_need=20
debug1: kex: ecdh-sha2-nistp256 need=20 dh_need=20
debug1: sending SSH2_MSG_KEX_ECDH_INIT
debug1: expecting SSH2_MSG_KEX_ECDH_REPLY
debug1: Server host key: RSA 16:27:ac:a5:76:28:2d:36:63:1b:56:4d:eb:df:a6:48
debug3: load_hostkeys: loading entries for host "github.com" from file "/home/ec2-user/.ssh/known_hosts"
debug3: load_hostkeys: found key type RSA in file /home/ec2-user/.ssh/known_hosts:1
debug3: load_hostkeys: loaded 1 keys
debug3: load_hostkeys: loading entries for host "140.82.118.3" from file "/home/ec2-user/.ssh/known_hosts"
debug3: load_hostkeys: loaded 0 keys
debug1: Host 'github.com' is known and matches the RSA host key.
debug1: Found key in /home/ec2-user/.ssh/known_hosts:1
Warning: Permanently added the RSA host key for IP address '140.82.118.3' to the list of known hosts.
debug1: ssh_rsa_verify: signature correct
debug2: kex_derive_keys
debug2: set_newkeys: mode 1
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
debug2: set_newkeys: mode 0
debug1: SSH2_MSG_NEWKEYS received
debug1: SSH2_MSG_SERVICE_REQUEST sent
debug2: service_accept: ssh-userauth
```

```
debug1: SSH2_MSG_SERVICE_ACCEPT received
debug2: key: /home/ec2-user/.ssh/id_rsa ((nil)),
debug2: key: /home/ec2-user/.ssh/id_dsa ((nil)),
debug2: key: /home/ec2-user/.ssh/id_ecdsa ((nil)),
debug2: key: /home/ec2-user/.ssh/id_ed25519 ((nil)),
debug1: Authentications that can continue: publickey
debug3: start over, passed a different list publickey
debug3: preferred gssapi-keyex,gssapi-with-mic,publickey,keyboard-interactive,password
debug3: authmethod_lookup publickey
debug3: remaining preferred: keyboard-interactive,password
debug3: authmethod_is_enabled publickey
debug1: Next authentication method: publickey
debug1: Trying private key: /home/ec2-user/.ssh/id_rsa
debug3: no such identity: /home/ec2-user/.ssh/id_rsa: No such file or directory
debug1: Trying private key: /home/ec2-user/.ssh/id_dsa
debug3: no such identity: /home/ec2-user/.ssh/id_dsa: No such file or directory
debug1: Trying private key: /home/ec2-user/.ssh/id_ecdsa
debug3: no such identity: /home/ec2-user/.ssh/id_ecdsa: No such file or directory
debug1: Trying private key: /home/ec2-user/.ssh/id_ed25519
debug3: no such identity: /home/ec2-user/.ssh/id_ed25519: No such file or directory
debug2: we did not send a packet, disable method
debug1: No more authentication methods to try.
Permission denied (publickey).
```

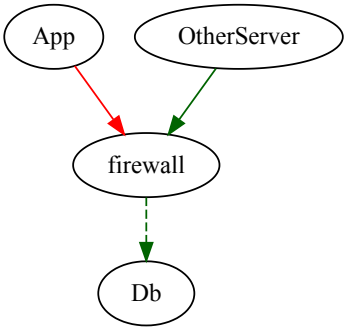
```
debug2: key: /home/ec2-user/.ssh/id_rsa ((nil)),
debug2: key: /home/ec2-user/.ssh/id_dsa ((nil)),
debug2: key: /home/ec2-user/.ssh/id_ecdsa ((nil)),
debug2: key: /home/ec2-user/.ssh/id_ed25519
```

Note that the look up policy can be influenced by `~/.ssh/config`
In the worst case scenario, you have plenty of available keys and ssh try them all and then server bans you for too much trials. 😟

My app can't connect to the db

Try as if in the network of your app, ideally on the same host.

The firewall rules, might allow one host and not another.



Let's enumerate some possible causes:

- firewall denying access? vlan segregation ?
- running on non-default port ?
- you don't use the correct url (host, port,...) ?
- you don't use the correct credentials ?
- no more storage available ? lack of disk space can lead to strange behavior
- server is really down ?

Tools to diagnose if the error message isn't that clear

Try with a command line tool

if the command line tool is available : psql, mysql,... or mssql-cli

```
mssql-cli -P xxxxx -U admin -S dbinstancename.environment.eu-central-1.rds.amazonaws.com
Error message: A network-related or instance-specific error occurred while establishing a connection to SQL Server. The server was not found or was not
```

Try with telnet

```
telnet dbinstancename.environment.eu-central-1.rds.amazonaws.com 1433
Trying 18.123.456.12...
Connected to ec2-18-123-456-12.eu-central-1.compute.amazonaws.com.
Escape character is '^]'.

^C^ZConnection closed by foreign host.
```

Try with netcat

telnet is often not installed by default, may be you have netcat

```
nc -v dbinstancename.environment.eu-central-1.rds.amazonaws.com 1443
nc: getaddrinfo: Name or service not known
```

nmap

self checking

```
nmap -sV github.com

Starting Nmap 7.01 ( https://nmap.org ) at 2019-07-08 21:26 CEST
Nmap scan report for github.com (140.82.118.4)
Host is up (0.039s latency).
rDNS record for 140.82.118.4: lb-140-82-118-4-ams.github.com
Not shown: 996 filtered ports
PORT     STATE SERVICE   VERSION
22/tcp   open  ssh       (protocol 2.0)
80/tcp   open  http
443/tcp  open  ssl/https GitHub.com
9418/tcp open  git?
3 services unrecognized despite returning data. If you know the service/version, please submit the following fingerprints at https://nmap.org/cgi-bin/s
```

You probably don't want to expose your db, redis, mongo,... server directly from the internet
nmap help

Making the web secure, one unit test at a time

http://gauntlt.org/

https://github.com/gauntlt/gauntlt/blob/master/examples/nmap/simple.attack

```
@slow
Feature: simple nmap attack (sanity check)

  Background:
    Given "nmap" is installed
    And the following profile:
      | name     | value      |
      | hostname | scanme.nmap.org |

  Scenario: Verify server is available on standard web ports
    When I launch an "nmap" attack with:
      """
      nmap -p 80,443 <hostname>
      """
    Then the output should match /80.tcp\s+open/
    And the output should not match:
      """
      443/tcp\s+open
      """
```

wireshark

Uptime monitoring tools

- https://www.uptrends.com/tools/uptime
- https://www.pingdom.com/product/uptime-monitoring/

WIP : Sample scenarios

- network
  - dns : no dns, ipv6, sometimes ok, sometimes not
  - tools
    - dig
    - pingdom (multisite)
- infra
  - diskspace : no space left on device
  - oom killer
  - ntpd ?
  - tools
    - df, du, iotop,...
- browser
  - ad blocker : /ad, /analytics,...
  - mixed content
  - csp
  - cors
- loadbalancer
  - bad gateway
  - http timeout
  - ddos, self-ddos
  - slow clients
  - heavy clients
  - queueing theory
- app
  - deadlock
    - lock
    - pool exhaustion (threads, db connections,...)
    - the importance of timeouts
  - slow
    - warm up
    - cache miss
    - verbose log level, synchronous, non buffered
    - regexp
  - infinite loop
    - cpu
  - garbage collection
    - pauses
    - plenty of object allocations
    - memory leaks
    - out of memory
  - strange bugs
    - fat client deployed from a shared drive
    - ftp active
  - zip/xml/... bomb
  - workers
    - paging
    - retries (becareful of too much retries)
  - tools :
    - thread dumps : kill -3
    - memory
    - application performance monitoring

- db
  - prepared statement
  - select n + 1
  - select * from db
  - orm vs sql
  - slow queries
  - non tuned memory/storage settings
  - tools :
    - explain plan
    - pgtop, mytop, db2top,... or their poor-man equivalent sql
- external services
  - degrade/fail gracefully