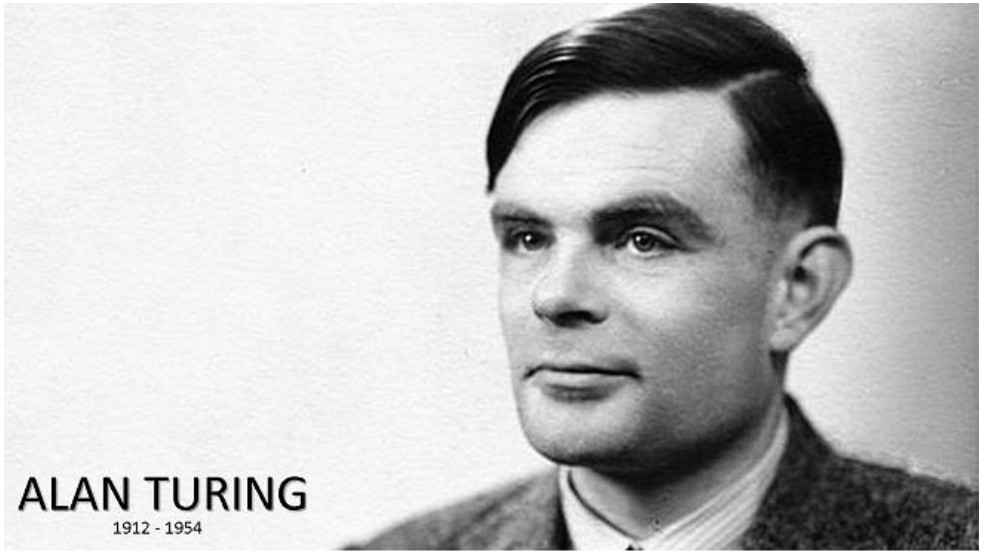
INFRASTRUCTURE AS CODE

WHAT IS IAC?



BEFORE THAT, SOME BASIC CONCEPTS

- Can the system perform logic?
- It is capable of iterate?
- And to expand memory?

SOME (UNINTENTIONAL) TURING COMPLETE SYSTEMS

- Microsoft excell
- C++ templates
- Power point
- Minesweeper
- Minecraft
- Magic: The Gathering

BEFORE THAT, SOME BASIC CONCEPTS

```
process {
  executor = 'sge'

  mapping.clusterOptions  = "-P project -pe omp 16 -l mem_total=94G"
  counting.clusterOptions = "-P project -pe omp 8  -l mem_total=16G"

  indir  = "/Users/pipeliner/pipelines/toy_data/rna-seq"
  outdir = "/Users/pipeliner/pipelines/rna-seq-results"
  fasta  = "${params.indir}/genome_reference.fa"
  gtf    = "${params.indir}/genome_annotation.gtf"

  // General pipeline parameters
  paired = true
  aligner = "hisat"
  quantifier = "featurecounts"

  index.use_existing = true
  index.path = "${params.indir}/alignment_indices/hisat_index/index/part"

  // Process-specific parameters
  feature_counts.cpus   = 8
  feature_counts.type   = "exon"
  feature_counts.id     = "gene_id"
  feature_counts.xargs  = ""
  feature_counts.ainj   = ""
}
```

BEFORE THAT, SOME BASIC CONCEPTS

Do we, humans, treat the same way config files and code files?

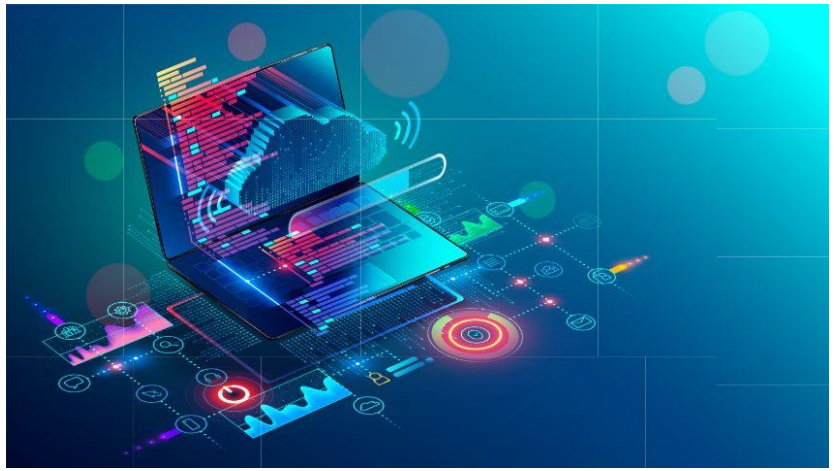
Are config files different from programs?

BEFORE THAT, SOME BASIC CONCEPTS

Software does everything code does... and more!

- Asserting libraries
- Integration testing
- Cross compiling
- Debugging
- Logging
- Versioning
- Refactoring
- Upgrading (Downgrading!)
- Flow control
- Linting
- Syntax control
- Observability
- Hinting, User experience
- Repeatability
- Compatibility
- Composability
- Inheritance
- Polymorphism

ANALYSIS OF BASIC CONCEPTS

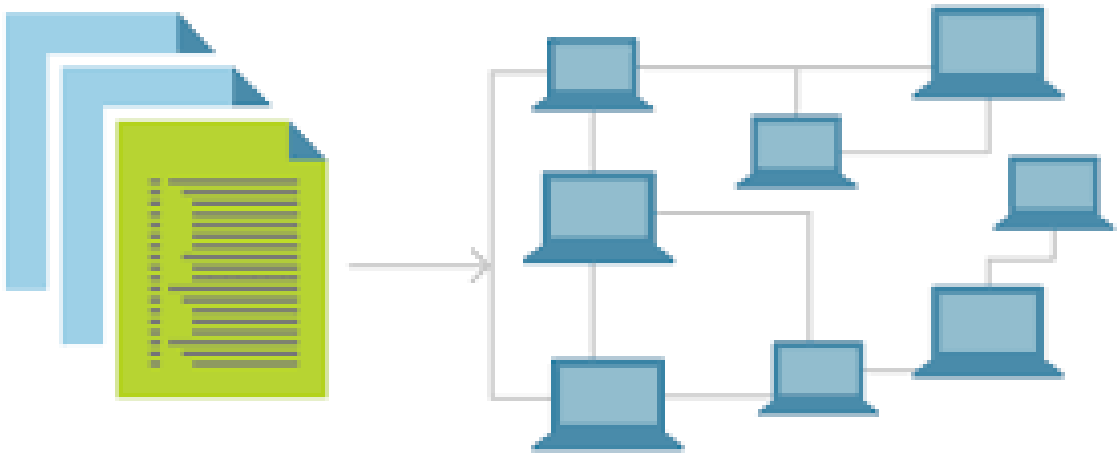


Example: imagine you have to configure a server and you are in a Turing Complete use case

```
process {
  executor = 'sge'

  mapping.clusterOptions  = "-P project -pe omp 16 -l mem_total=94G"
  counting.clusterOptions = "-P project -pe omp 8  -l mem_total=16G"

  indir  = "/Users/pipeliner/pipelines/toy_data/rna-seq"
  outdir = "/Users/pipeliner/pipelines/rna-seq-results"
  fasta  = "${params.indir}/genome_reference.fa"
  gtf    = "${params.indir}/genome_annotation.gtf"

  // General pipeline parameters
  paired = true
  aligner = "hisat"
  quantifier = "featurecounts"

  index.use_existing = true
  index.path = "${params.indir}/alignment_indices/hisat_index/index/part"

  // Process-specific parameters
  feature_counts.cpus   = 8
  feature_counts.type   = "exon"
  feature_counts.id     = "gene_id"
  feature_counts.xargs  = ""
  feature_counts.ainj   = ""
}
```

CONFIGURATE INFRASTRUCTURE

- Nowadays we are already using config files to manage infrastructures
- Config file: hard coding
- Use software: advantages everywhere



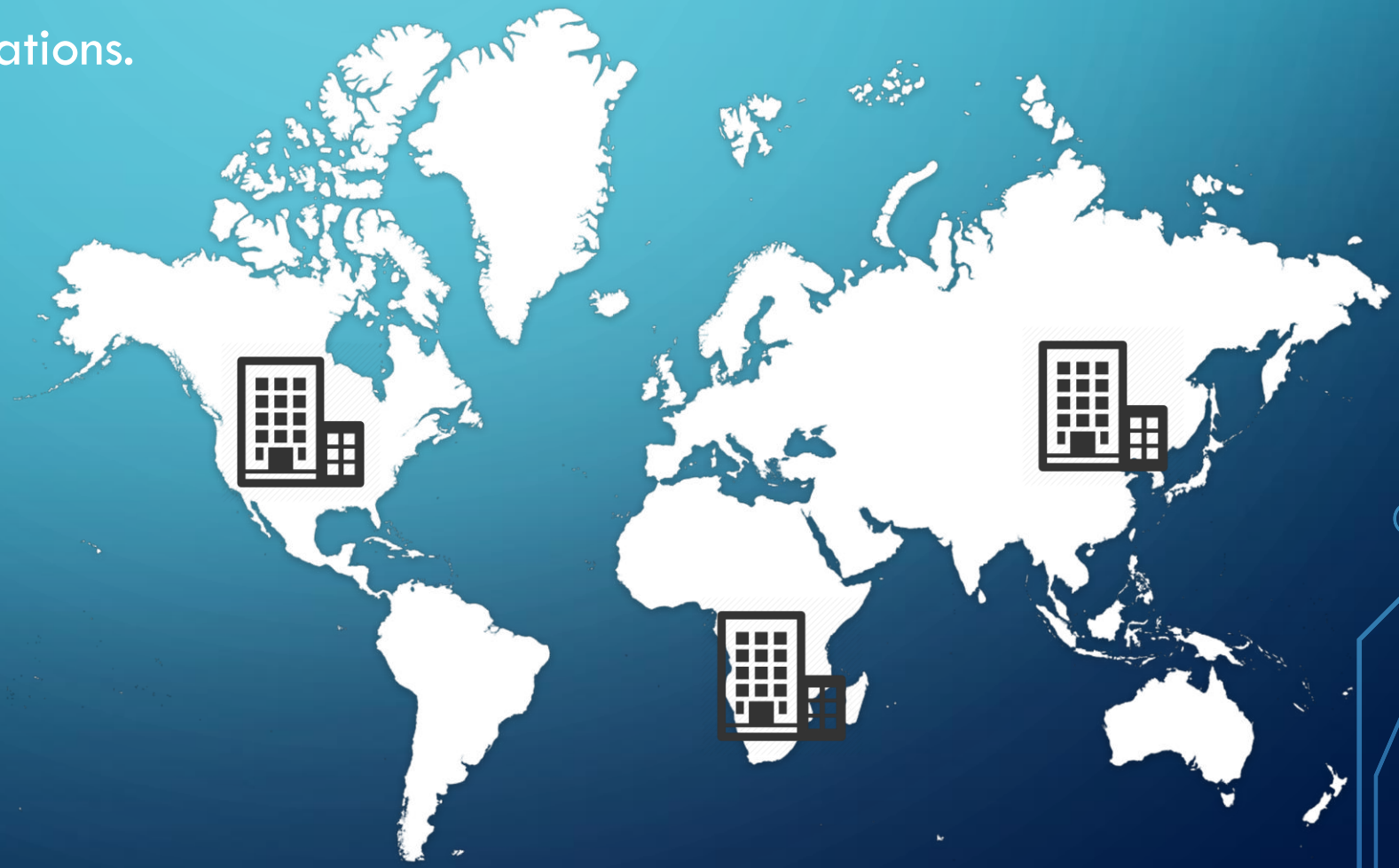
SO, WHAT IS IAC OR IAS?

Avoid the use of config files to manage the configuration of infrastructure and instead use code or software

WHAT ARE THE COMMON USES OF IAC?

DECENTRALIZED COMPANIES

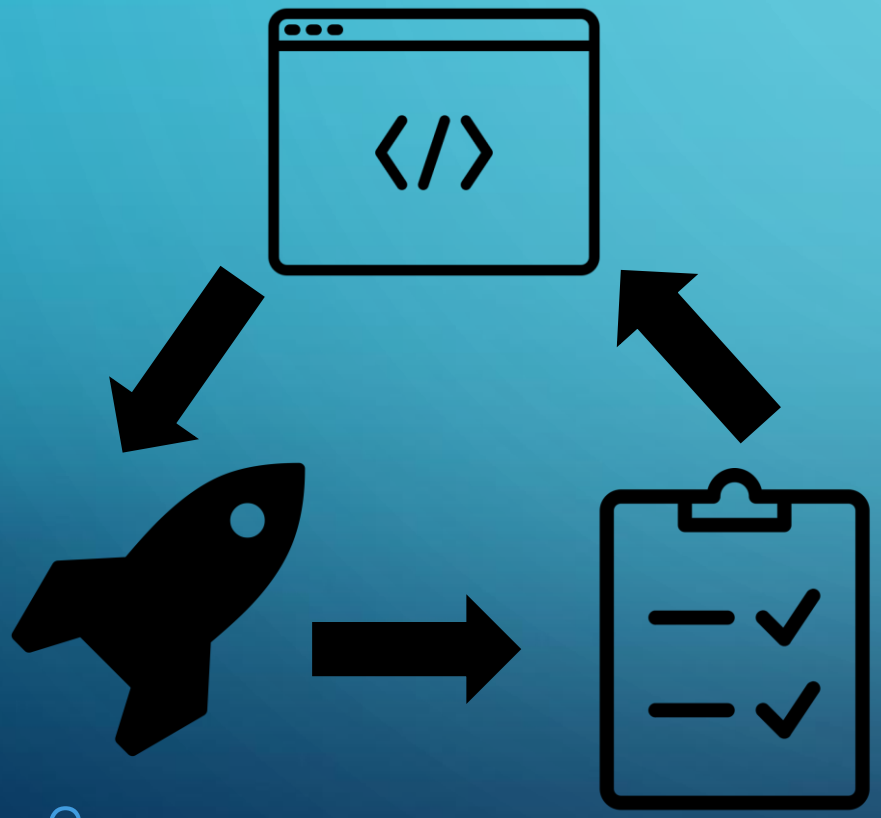
- Very common: companies with several hq
- Hq in different regions == compatibility issues
- IaC unifies system configurations.



SERVER DEPLOYMENT

- Need of contuous services
- Possible scenario: machine break down
- IaC allows for faster deployment

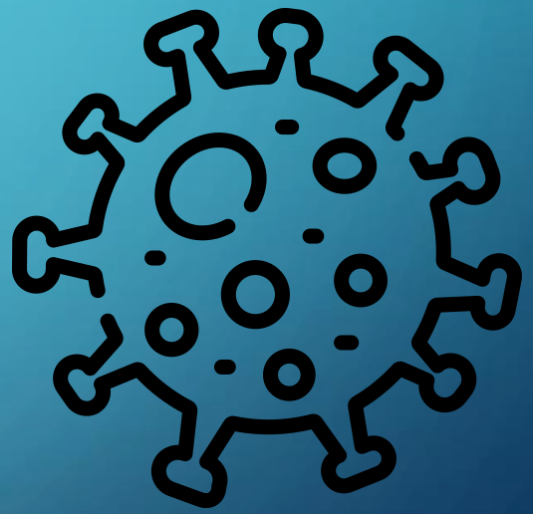
CONTINUOUS INTEGRATION

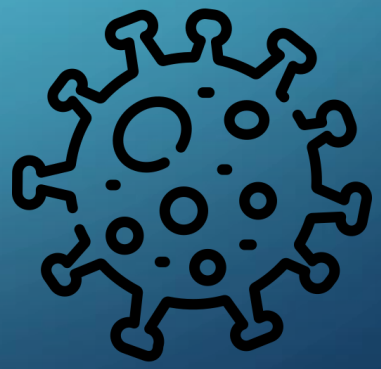


- Deploying only at the end of development = problems
- Deploying manually and periodically = time consuming
- IaC = solution

SECURITY

- IaC template incident
- Vulnerability propagation
- Solution: Compliance as code



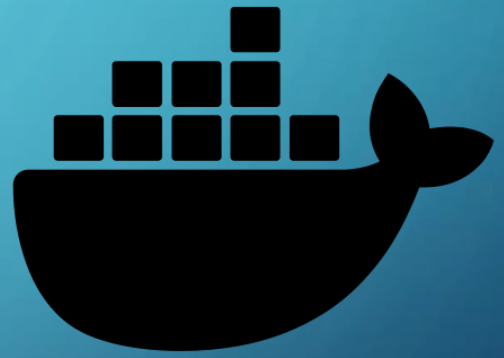


HAVE WE USED IAC AS STUDENTS?

Vagrant

vmware

VM



Docker

WHEN DO WE USE IAC?

SHOULD I USE IAC?

SIMPLICI

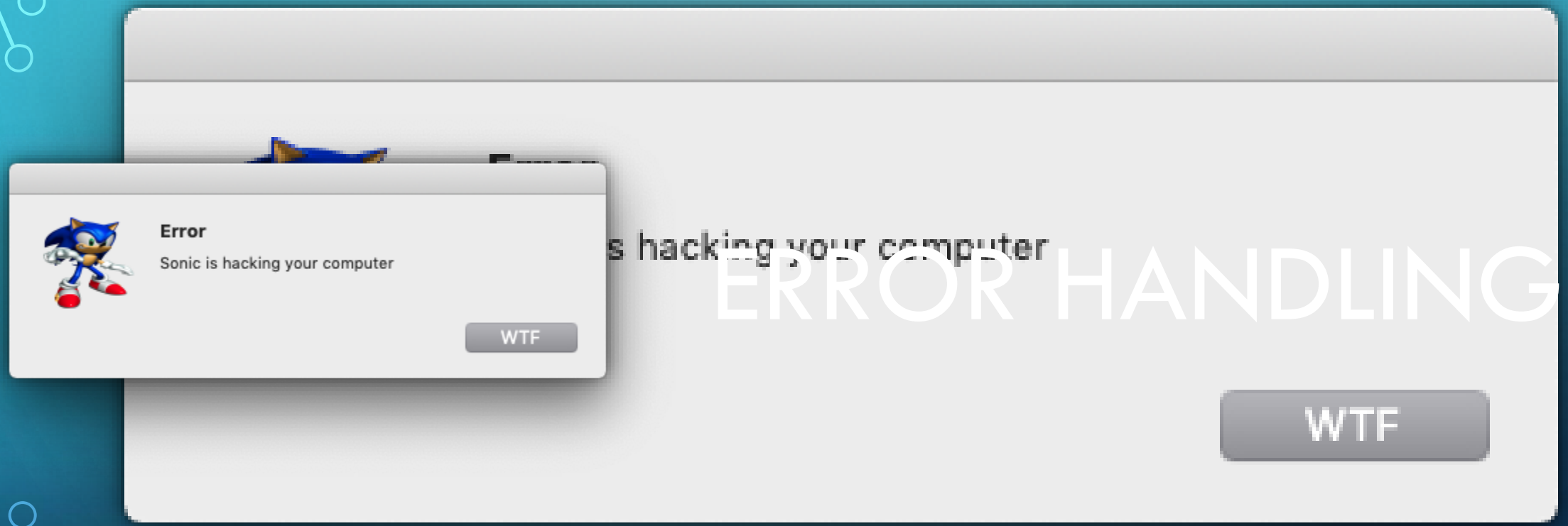
ABILITY

SHAREABILITY

TY

STAB

ERROR HANDLING

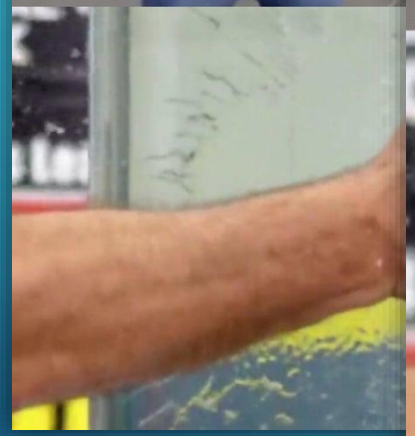


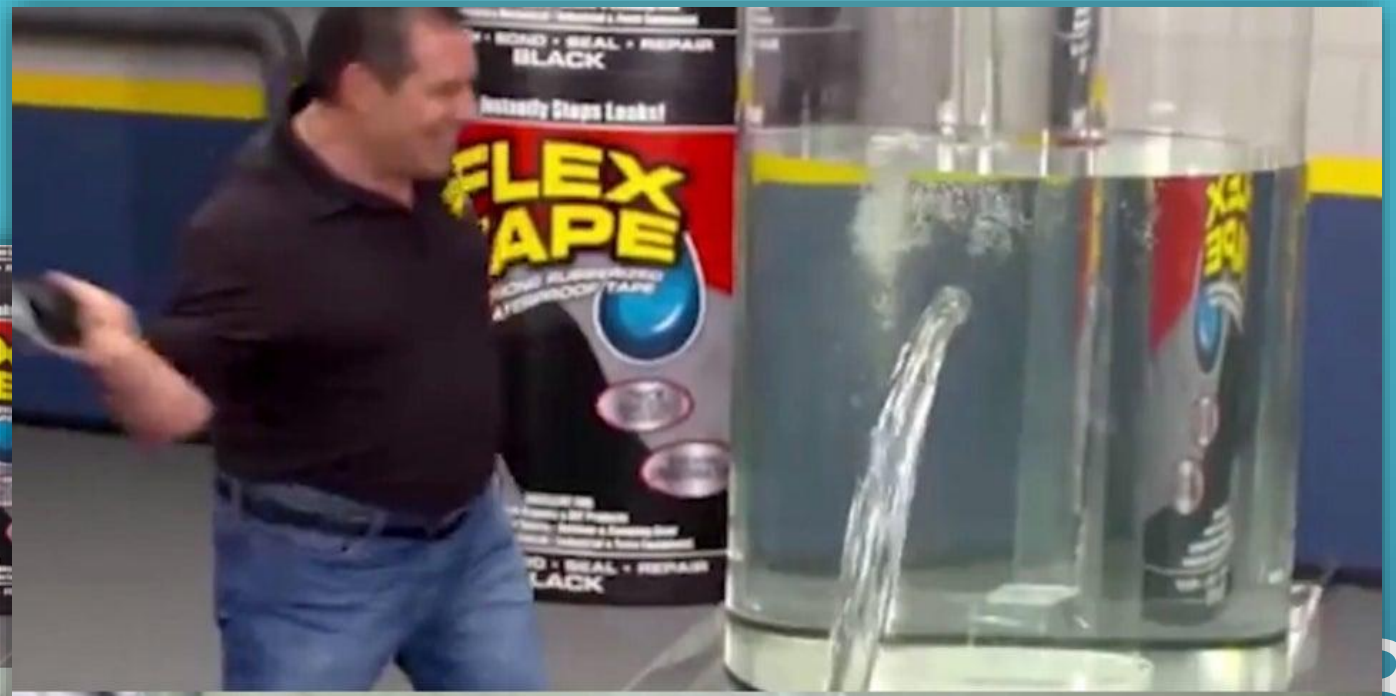
SCALA


My IaC code with different configurations
My IaC code
with different
s

BILITY


BOND • SEAL • REPAIR
BLACK

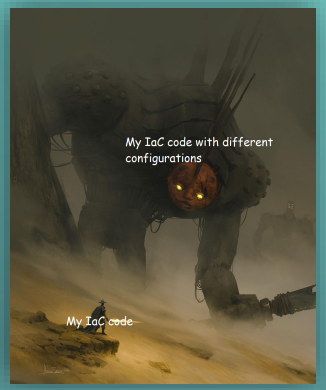



BOND • SEAL • REPAIR
BLACK
FLEX
APE
BOND • SEAL • REPAIR
BLACK

SIMPLICITY

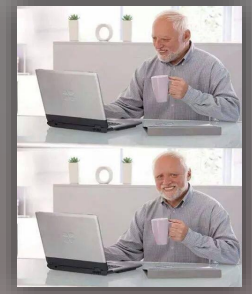
SHAREABILITY

STABILITY

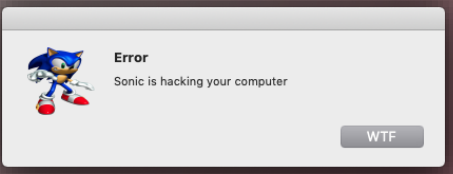


SCALABILITY

UNDERSTANDABILITY



LEARNABILITY



ERROR HANDLING

MAINTAINABILITY