

# Infrastructure-as-Code with Pulumi

Better than all the others (like Ansible)?

Jonas Hecht | Senior IT-Nerd | [github.com/jonashackt](https://github.com/jonashackt)

 @jonashackt

<https://jonashackt.io>



[jonashackt.io](http://jonashackt.io)



4+1



FACHHOCHSCHULE  
ERFURT UNIVERSITY  
OF APPLIED SCIENCES  
Angewandte  
Informatik



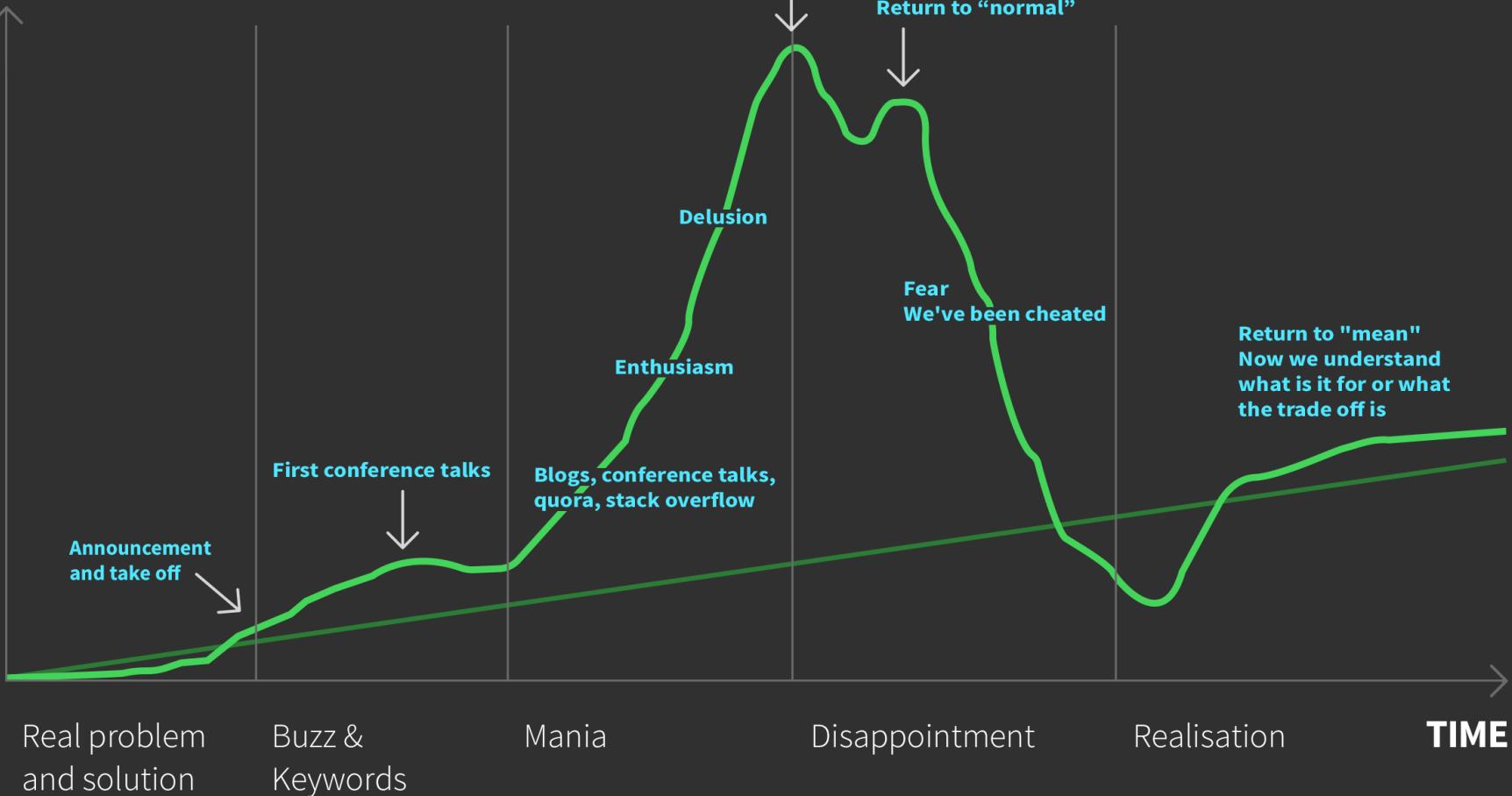
Bauhaus-  
Universität  
Weimar



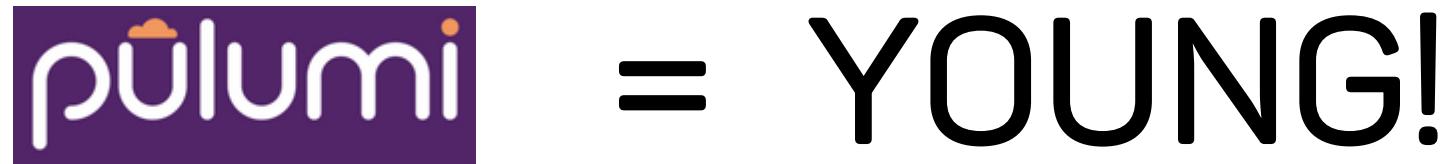
Our industry is like...



# HYPE

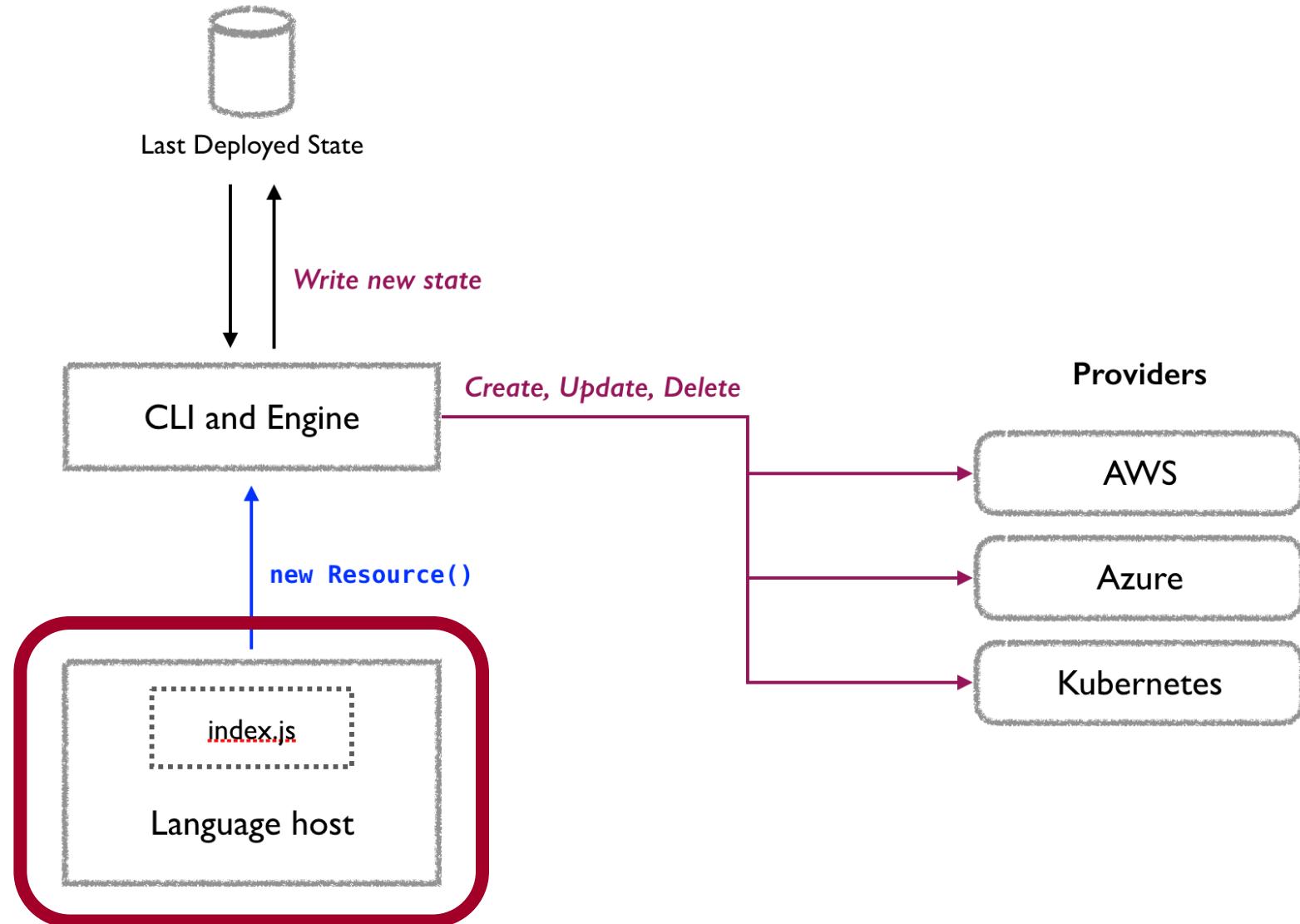


Pulumi?



published June 2018

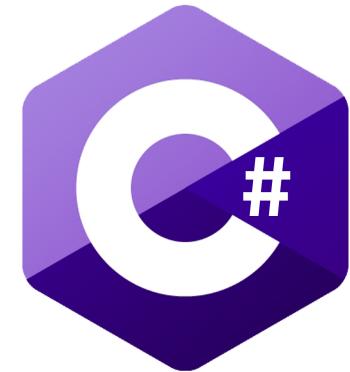
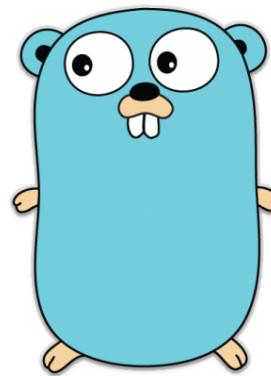
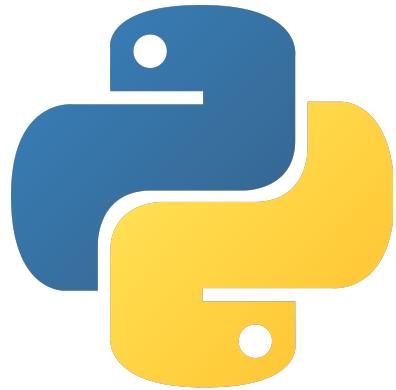
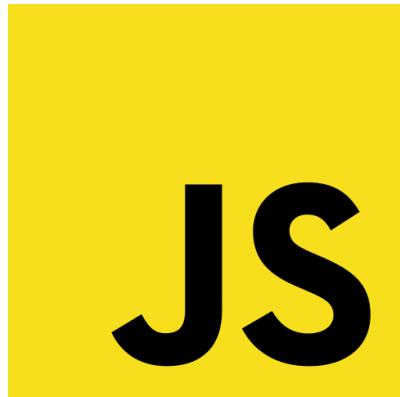
1.0 September 2019



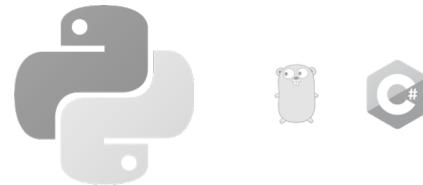
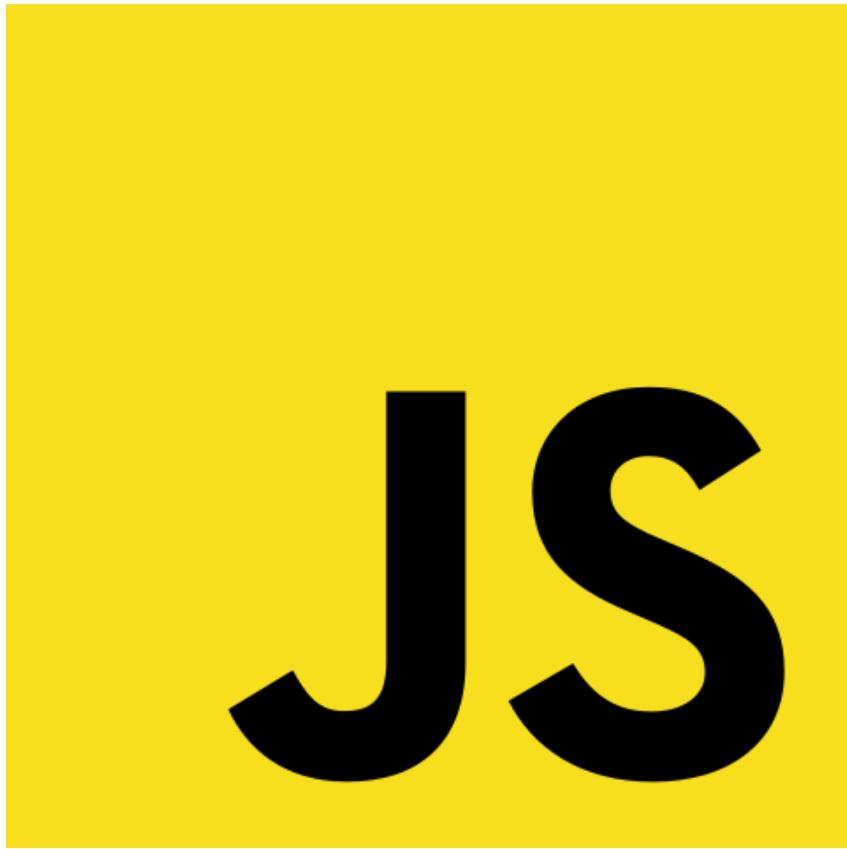
“

Pulumi is language agnostic to support multiple programming languages at the same time

[pulumi.com/docs/intro/languages](https://pulumi.com/docs/intro/languages)



[pulumi.com/docs/intro/languages](https://pulumi.com/docs/intro/languages)



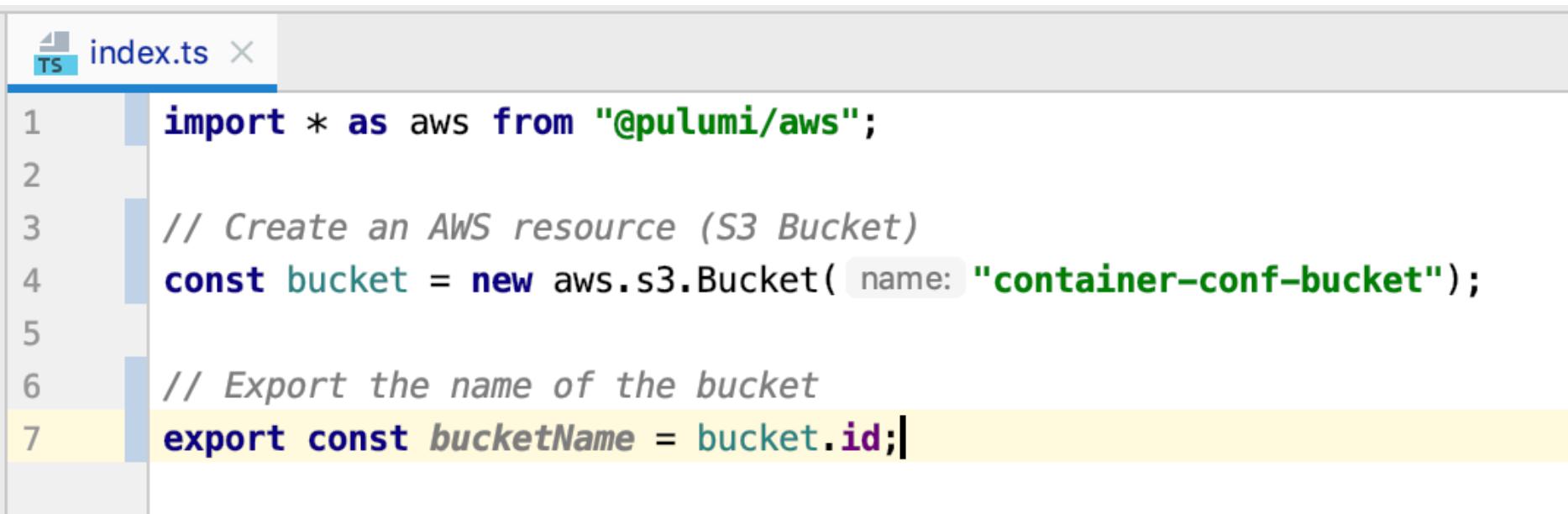
API reference: [pulumi.com/docs/reference/pkg](https://pulumi.com/docs/reference/pkg)

Examples: [github.com/pulumi/examples](https://github.com/pulumi/examples)

|   |                             |
|---|-----------------------------|
|  <a href="#">.gitignore</a>        |                             |
|  <a href="#">.travis.yml</a>       |                             |
|  <a href="#">LICENSE</a>           |                             |
|  <a href="#">Pulumi.dev.yaml</a>   | Pulumi Stack definition     |
|  <a href="#">Pulumi.yaml</a>       | Pulumi Project config       |
|  <a href="#">README.md</a>         |                             |
|  <a href="#">index.ts</a>          | Pulumi Program (TypeScript) |
|  <a href="#">package-lock.json</a> |                             |
|  <a href="#">package.json</a>      | Node.js Package Management  |
|  <a href="#">renovate.json</a>    |                             |
|  <a href="#">tsconfig.json</a>   | TypeScript config           |

[github.com/jonashackt/pulumi-typescript-aws-fargate](https://github.com/jonashackt/pulumi-typescript-aws-fargate)

# Pulumi program example



The screenshot shows a code editor window with a tab labeled "index.ts". The code itself is as follows:

```
1 import * as aws from "@pulumi/aws";
2
3 // Create an AWS resource (S3 Bucket)
4 const bucket = new aws.s3.Bucket({ name: "container-conf-bucket" });
5
6 // Export the name of the bucket
7 export const bucketName = bucket.id;
```

The code uses the `@pulumi/aws` package to create an S3 Bucket named "container-conf-bucket". It then exports the bucket's ID as `bucketName`. The code editor highlights the file extension ".ts" and the word "TS" in the tab bar.

# Create projects



```
~/dev > mkdir containerconf && cd containerconf  
~/dev/containerconf > pulumi new aws-typescript
```

# Multicloud?



Google Cloud Platform



kubernetes

[pulumi.com/docs/intro/cloud-providers](https://pulumi.com/docs/intro/cloud-providers)

# aws



Google Cloud Platform

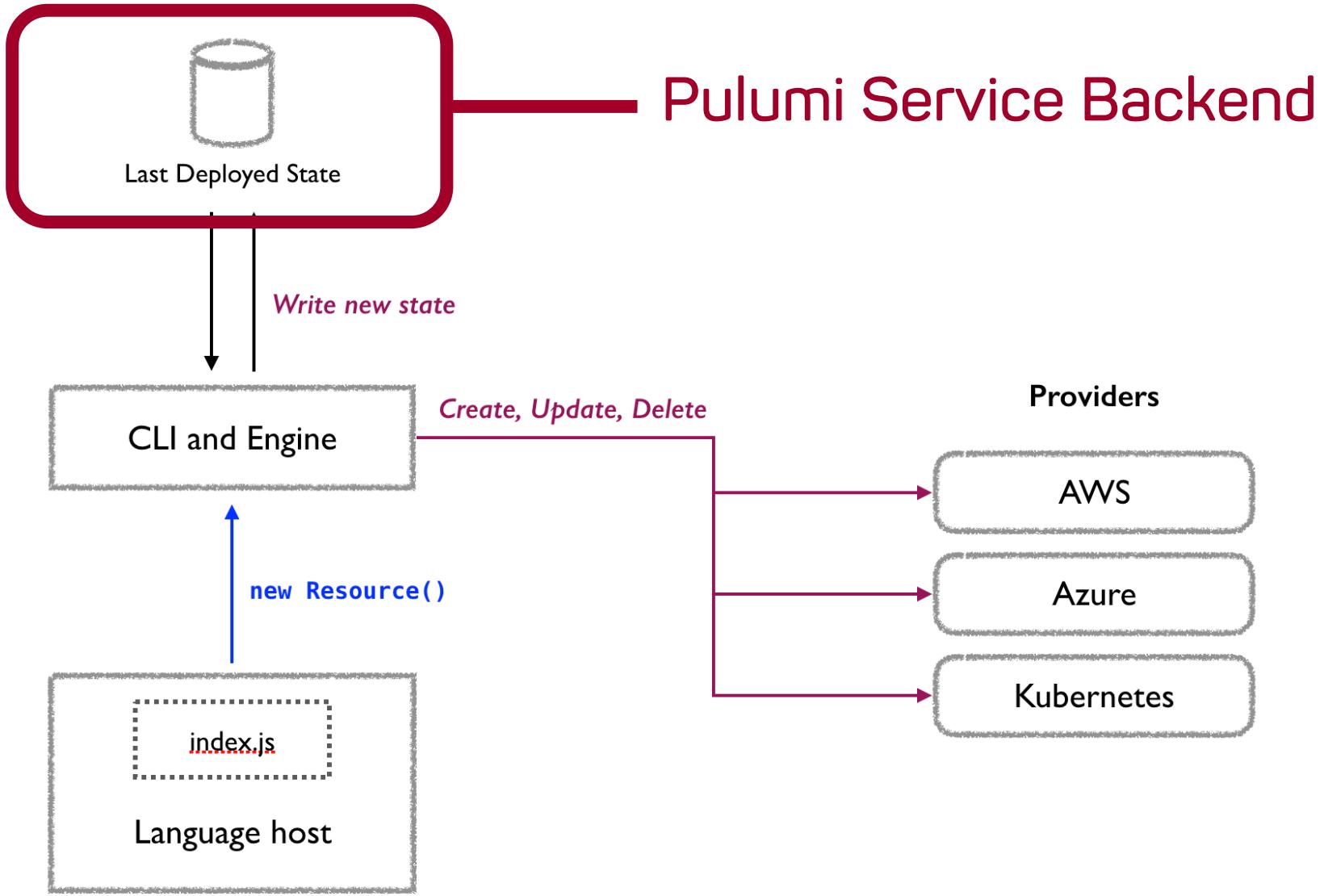


# kubernetes

[github.com/pulumi/examples](https://github.com/pulumi/examples)

# State

State  
is stored in the Pulumi  
Service Backend



# Service Backend options

- ▶ [app.pulumi.com](https://app.pulumi.com)
- ▶ Self-managed backend
- ▶ on-premise [app.pulumi.com](https://app.pulumi.com)  
(Enterprise version only)

[pulumi.com/docs/intro/concepts/state/](https://pulumi.com/docs/intro/concepts/state/)

A **Stack** defines the **State**  
of a Pulumi project

Example: [app.pulumi.com/jonashackt](https://app.pulumi.com/jonashackt)

.gitignore

.travis.yml

LICENSE

Pulumi.dev.yaml

Pulumi.yaml

README.md

index.ts

package-lock.json

pulumi.yaml

renovate.json

tsconfig.json

Pulumi.dev.yaml

```
YML config:
  aws:region: eu-central-1
  pulumi-example-aws-python:dbPassword:
    secure: AAABAGRrYHRPA4g1SBcTC4iUea203KcLx3l+pBpY11PFtueLCUE=
```

# Pulumi.YourStackName.yaml

Resource Providers  
aka  
„Terraform Wrappers“

PRODUCTS

PRICING

DOCS

BLOG

WHY PULUMI

ABOUT

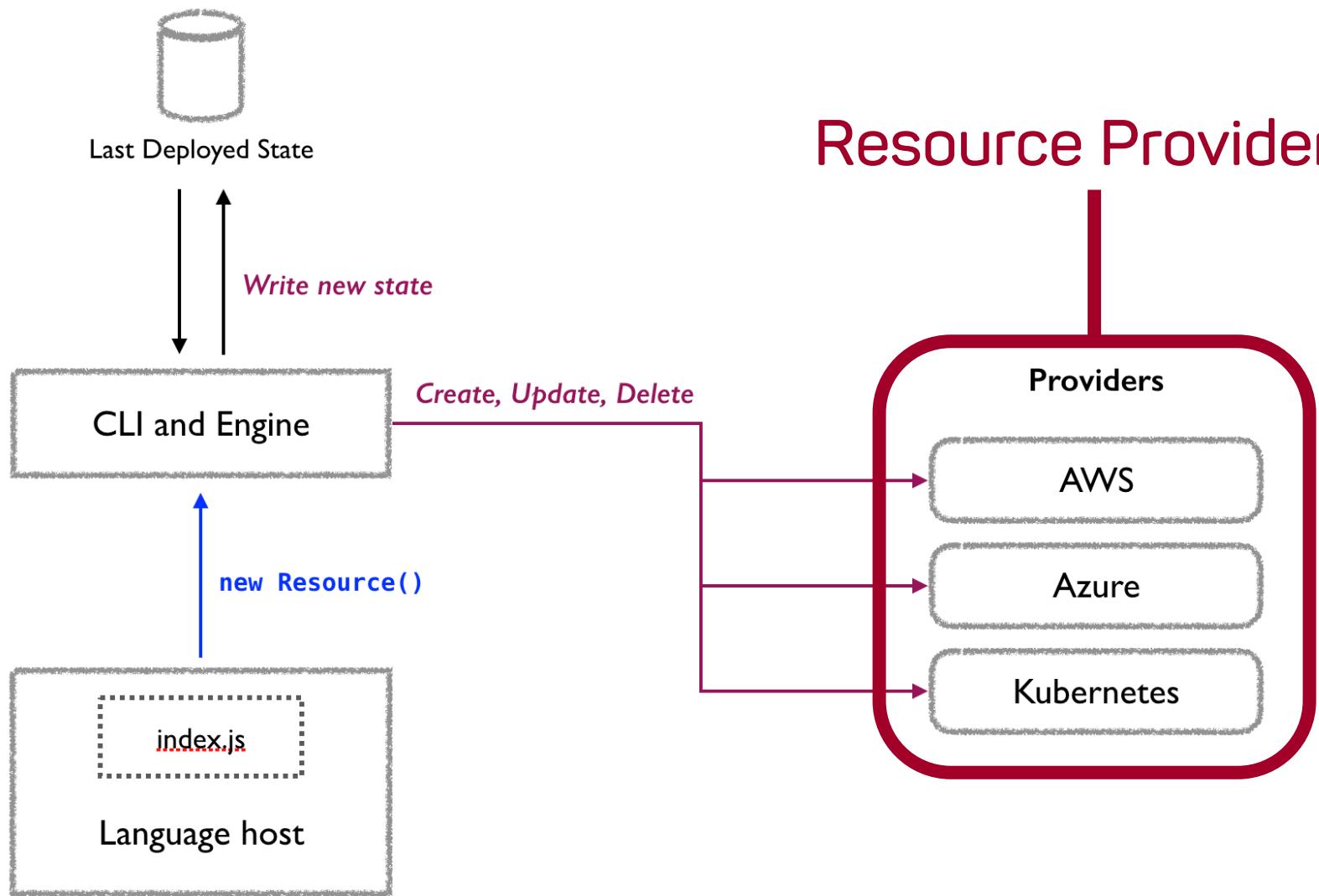
Docs / Reference / API / @pulumi/aws / elasticsearch

# Module elasticsearch

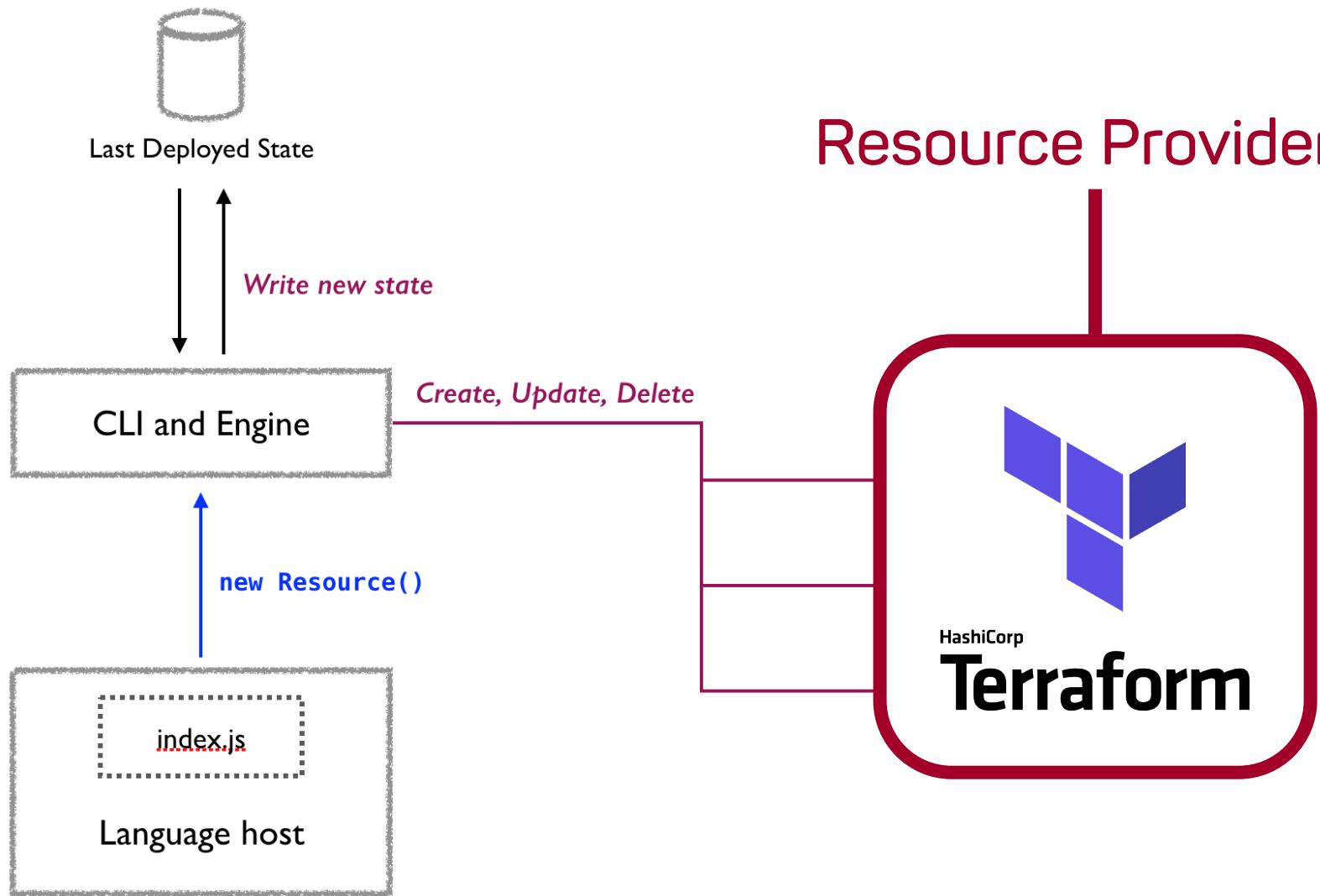
This provider is a derived work of the [Terraform Provider](#), distributed under [MPL 2.0](#). If you encounter a bug or missing feature, first check the [pulumi/pulumi-aws](#) repo; however, if that doesn't turn up anything, please consult the source [terraform-providers/terraform-provider-aws](#) repo.



# Resource Providers



# Resource Providers





= „Terraform  
Wrapper“

2019!

[pulumi.com/docs/intro/vs/terraform/#converting-from-terraform](https://pulumi.com/docs/intro/vs/terraform/#converting-from-terraform)

## M30 (December 2019)

- Custom update and delete hooks (e.g. provisioners)
- .NET Support
- Gated Deployments

## Backlog

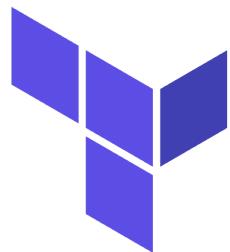
- [tf2pulumi support for Python](#)
- Complete support for Go
- Java Support
- High level libraries (like `awsx`) for Azure and GCP
- High level libraries (like `awsx`) for Python
- Overhaul `cloud` for true cross-cloud infrastructure definition
- **Native Pulumi providers for AWS/Azure/GCP**

[github.com/pulumi/pulumi/wiki/Roadmap](https://github.com/pulumi/pulumi/wiki/Roadmap)

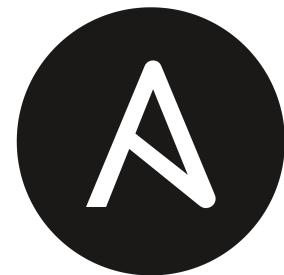
Pulumi vs. X



SALTSTACK



HashiCorp  
**Terraform**



ANSIBLE



CloudFormation



vs.



1. Academic comparison
2. Handle configuration drift
3. Current state of my infrastructure?
4. Tools shouldn't suck! (no master!, no agents!)
5. Able to do the job! (all major Clouds & on-premise)

# 1. Academic comparison

# Configuration Management



ANSIBLE

-----

# Provisioning



HashiCorp  
**Terraform**



ANSIBLE

“

Pulumi is fundamentally different  
than these tools and works great  
alongside them

[pulumi.com/docs/intro/vs/](https://pulumi.com/docs/intro/vs/)

# Configuration Management

# Provisioning



ANSIBLE

+



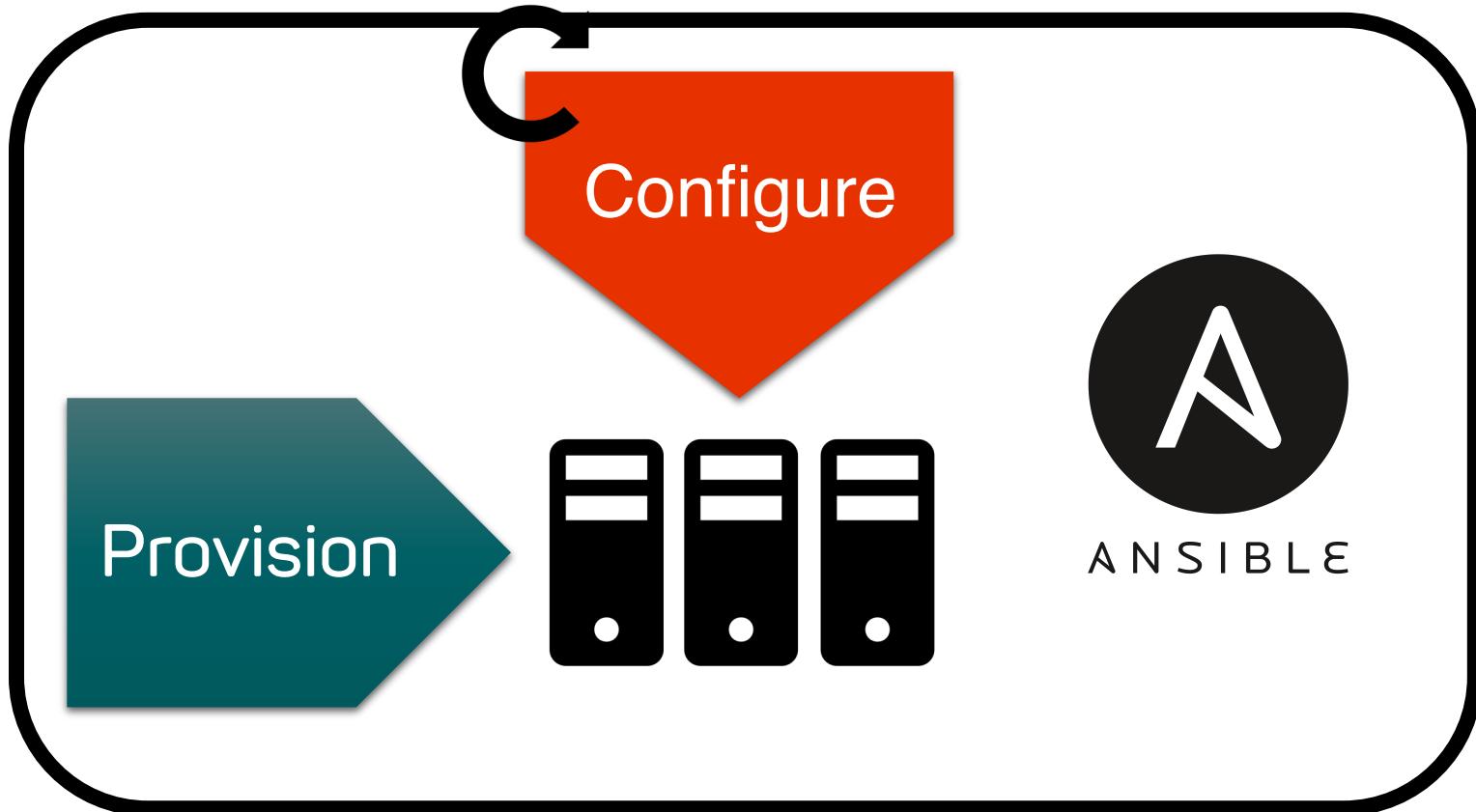
2. How do they handle configuration drift?

“

the state of the machine  
drifts from the baseline due to  
manual changes and updates

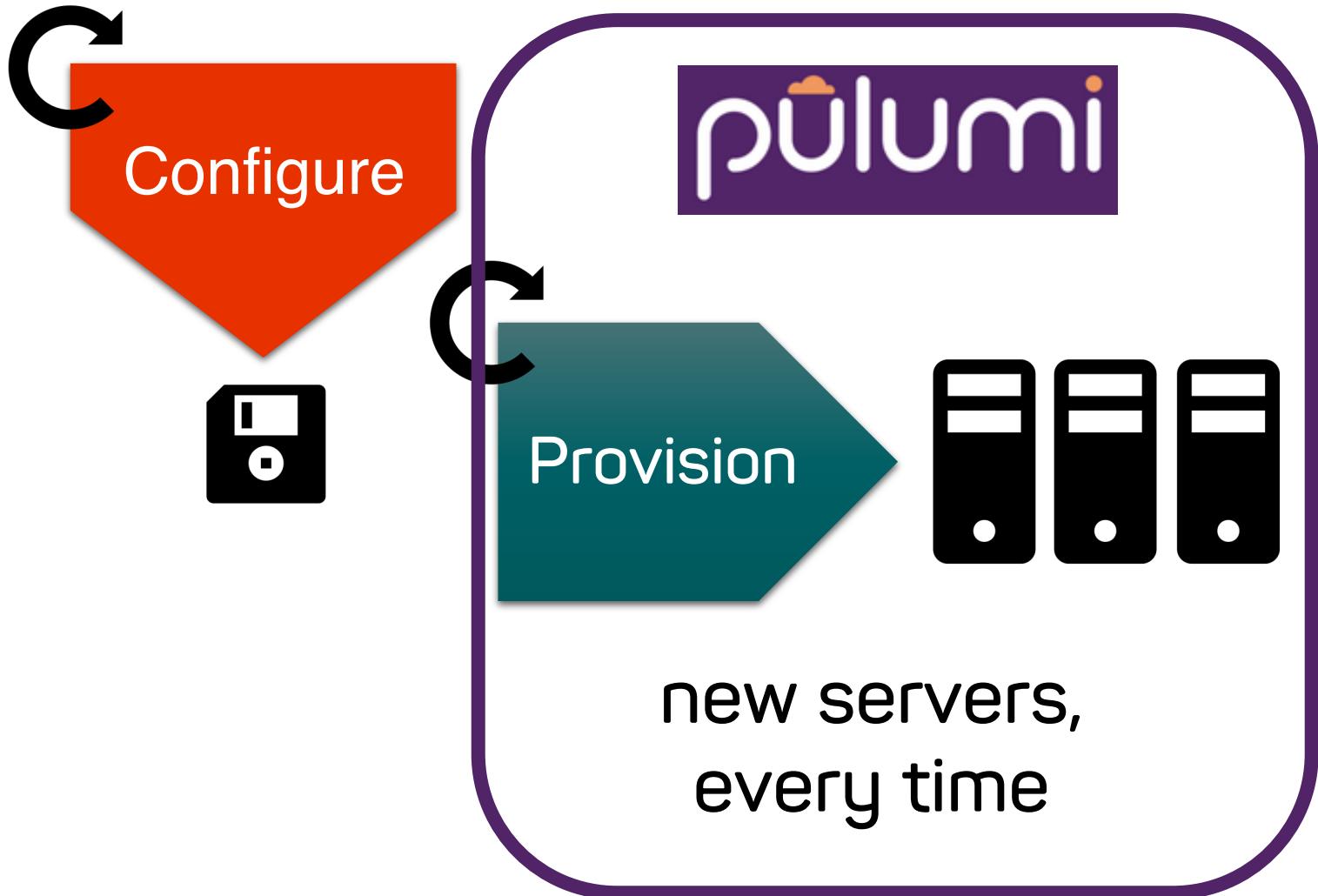
[shadow-soft.com/ansible-idempotency-configuration-drift](https://shadow-soft.com/ansible-idempotency-configuration-drift)

## Mutable Infrastructure



same servers,  
changed every time

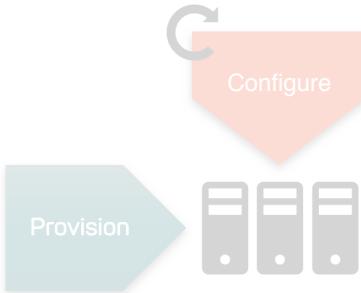
## Immutable Infrastructure



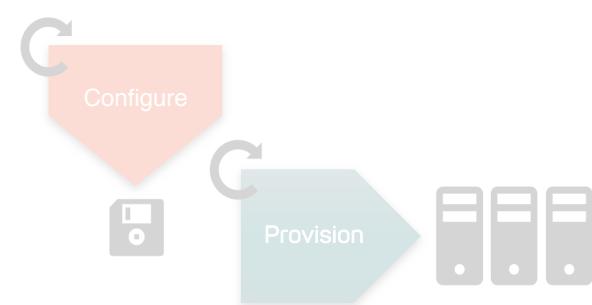
What if we don't allow „manual changes“?

Use software engineering practices  
like  
Continuous Integration  
for your infrastructure code!

## Mutable Infrastructure



## Immutable Infrastructure



Running our Infrastructure code in CI/CD pipelines, this comparison become less relevant!

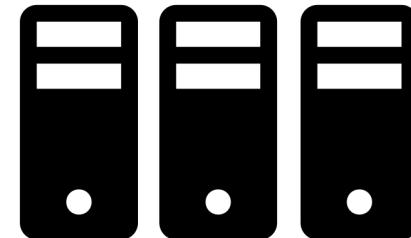


Demo time!

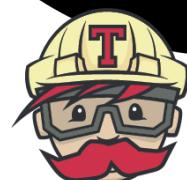
pulumi

Provision

Configure



Test



Travis CI

[github.com/jonashackt/pulumi-python-aws-ansible](https://github.com/jonashackt/pulumi-python-aws-ansible)

# 3. Current state of my infrastructure?

There's this 3rd comparison called Procedural vs.  
Declarative...

But remember the  
Continuous Integration  
thingy?

# 4. Tools shouldn't suck

(no master!, no agents!)

## master

- central place to see status of your infrastructure
- continuously enforce configuration in the background
- extra infrastructure!
- needs to be maintained
- client-2-master & master-2-servers communication needs ports & security

## no master

- 
- 
- 
- 
- 
- 
- 

Remember the  
Continuous Integration  
Thingy again?

master



SALTSTACK



puppet



CHEF



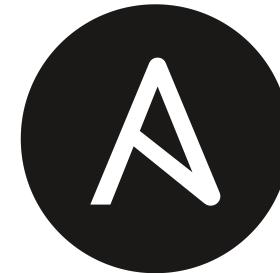
- - - - -



HashiCorp  
Terraform



CloudFormation



ANSIBLE



Agent needs to be  
installed on the  
server



-----

No Agent  
needed



# 5. Able to do the job!

(all major Clouds & on-premise)



ANSIBLE





**Ansible**  
5K Stacks



**Terraform**  
2.3K Stacks



**Pulumi**  
22 Stacks



Stacks

**5K**

Followers

**3.6K**

Votes

**1.2K**

Stacks

**2.3K**

Followers

**1.5K**

Votes

**253**

Stacks

**22**

Followers

**28**

Votes

**2**



484



1.3K



11.8K



153



103



3.8K



49



40.1K



17.3K



1h



19.4K



5.2K



1d



3.9K



200



7h

[stackshare.io/stackups/ansible-vs-terraform-vs-pulumi](https://stackshare.io/stackups/ansible-vs-terraform-vs-pulumi)

💪 google.com 😢

💪 stackoverflow.com 😢

💪 documentation 😢

💪 blog & articles count 😢

💪 community size 😢

💪 on-premise also 😢

💪 Both config mgt. &  
provisioning 😢

⌚ 48,130 commits



👤 4,738 contributors

⌚ 4,046 commits



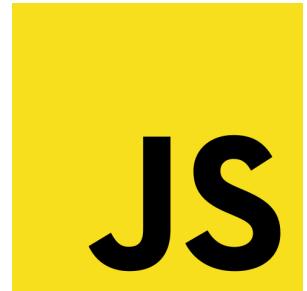
👤 51 contributors

# One more thing...





FOR AMAZON WEB SERVICES (AWS)



“

Crosswalk for AWS supports “day one” tasks, such as creating your initial container-based workloads using ECS/Fargate/EKS & serverless workloads (API Gateway / Lambda)

[pulumi.com/docs/guides/crosswalk/aws](https://pulumi.com/docs/guides/crosswalk/aws)

## APPLICATION DELIVERY

Go to production with containers and serverless.



Containers

Serverless

Continuous Delivery

## INFRASTRUCTURE PATTERNS

Well-Architected infrastructure for modern applications.



Security

Networking

Clusters

Monitoring

## FOUNDATION ALL OF AWS

Unopinionated infrastructure as code for all AWS resources and features.



↑ PRODUCTIVITY  
↓ CONTROL



pulumi  
crosswalk

FOR AMAZON WEB SERVICES (AWS)

JS

[github.com/jonashackt/pulumi-typescript-aws-fargate](https://github.com/jonashackt/pulumi-typescript-aws-fargate)

App: [github.com/jonashackt/spring-boot-vuejs](https://github.com/jonashackt/spring-boot-vuejs)

## M30 (December 2019)

- Custom update and delete hooks (e.g. provisioners)
- .NET Support
- Gated Deployments

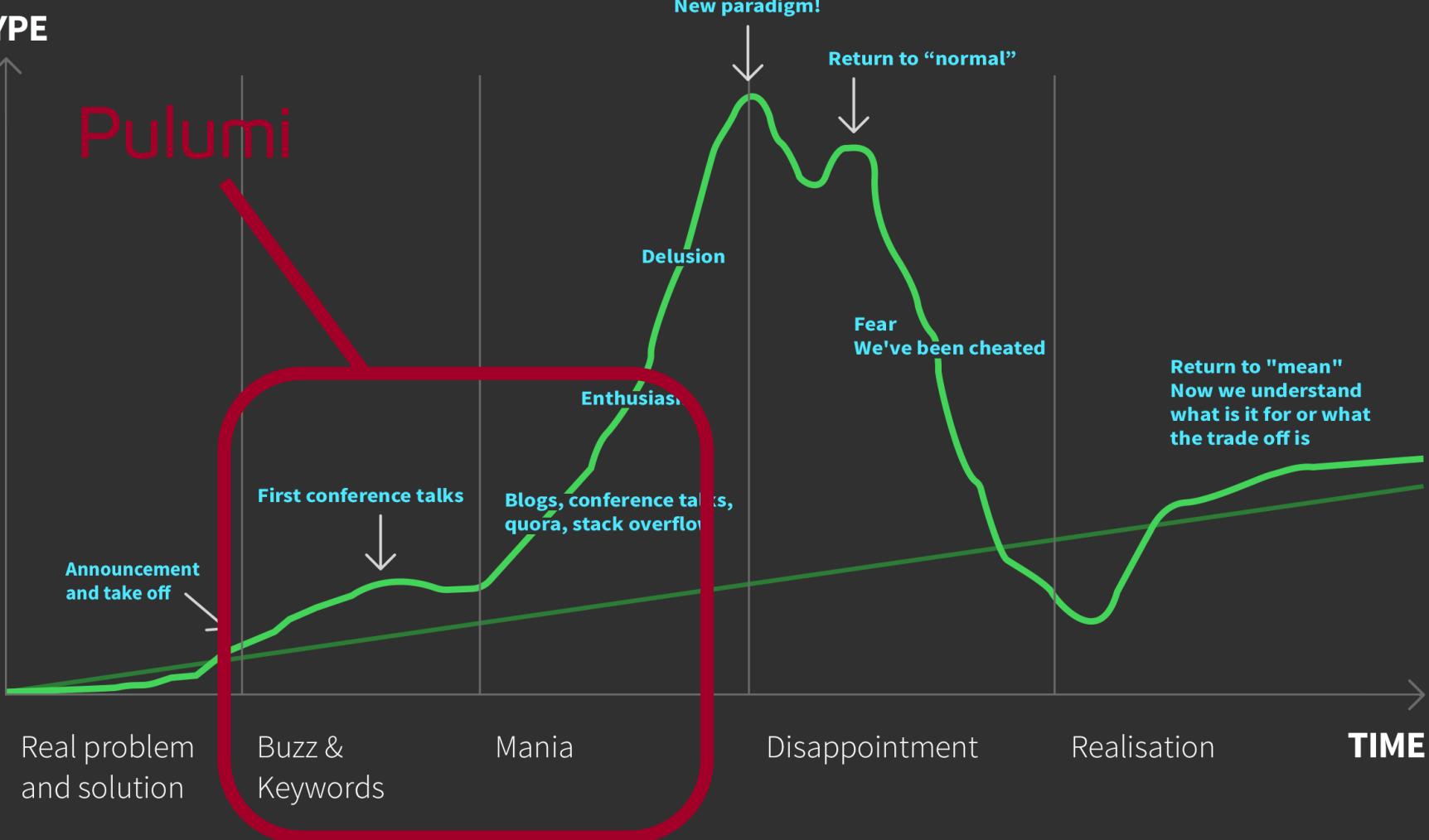
## Backlog

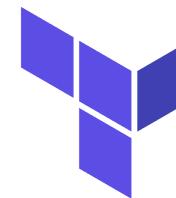
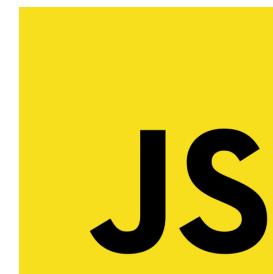
- `tf2pulumi` support for Python
- Complete support for Go
- Java Support
- High level libraries (like `awsx`) for Azure and GCP
- High level libraries (like `awsx`) for Python
- Overhaul `cloud` for true cross-cloud infrastructure definition
- Native Pulumi providers for AWS/Azure/GCP

[github.com/pulumi/pulumi/wiki/Roadmap](https://github.com/pulumi/pulumi/wiki/Roadmap)

# Summary

HYPE





HashiCorp  
**Terraform**



**pulumi**  
**crosswalk**  
FOR AMAZON WEB SERVICES (AWS)



# No matter what IaC tool you use...

Treat your infrastructure code  
**AS CODE!**

Use Testframeworks – or even do TDD!

Run Tests automatically - in your  
Continuous Integration Pipeline!

Run your infrastructure code frequently!  
(scheduled CI jobs)

Always aim for reproducible builds (aka  
dependency management)

Automatically update dependencies  
(e.g. renovatebot)



[github.com/jonashackt/pulumi-talk](https://github.com/jonashackt/pulumi-talk)

@jonashackt