Configuration and secret management in AWS

going native

Gonçalo Pestana - March 21, 2018

Configuration and secret management in AWS

Gonçalo Pestana **@gpestana**

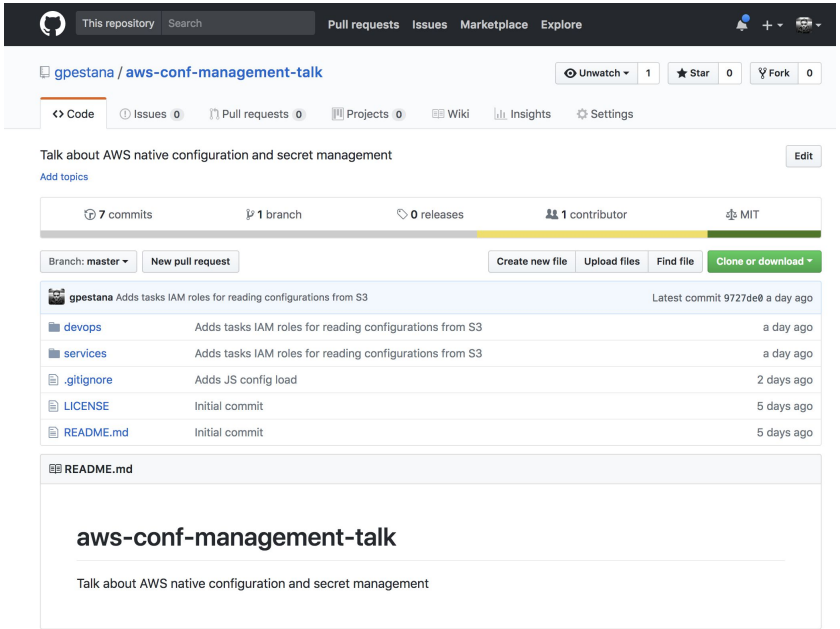
Senior developer at New Things Company

Configuration and secret management in AWS

Feedback, poll & discussion at **@gpestana**

Configuration and secret management in AWS

Slide deck, demo code and TF infra



Configuration and secret management in AWS

Going from vendor solution to **AWS native**

Config and secret management & AWS

Demo

Wrap up

- It is hard to maintain critical systems in production (e.g. configuration and secret management)

- AWS services can takes us far in config and secret management
  - S3, IAM, KMS, SNS, SQS, audit logs ...

- Many different architectures. Simplest one rely only on S3 and polling by config. consumers

- Configuration polling on application level OR parent-process level

Configuration and secret management

Hashicorp Consul

Service discovery

Configuration management (KV storage)

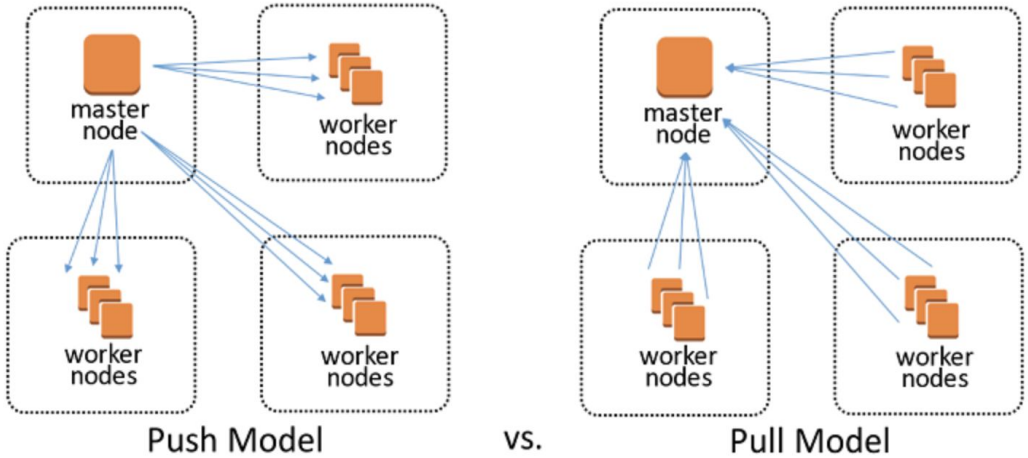
Hashicorp Vault

Stores and controls access tokens, passwords, certificates and secrets

Key revocation, key rolling and auditing

Configuration and secret management

Consul, Chef, Puppet, Ansible, Salt, …



source: https://aws.amazon.com/pt/answers/configuration-management/aws-infrastructure-configuration-management/

Going native - why?

"Consul is a highly available and distributed service which..."

Going native - why?

"Consul is a highly available and distributed service which..."

Deploying and managing distributed systems is hard

Vault and Consul are - still - critical points of failure that need caretaking

Cross platform/service (EC2 instances, lambda functions, ECS containers..)

Simplicity

Going native - why?

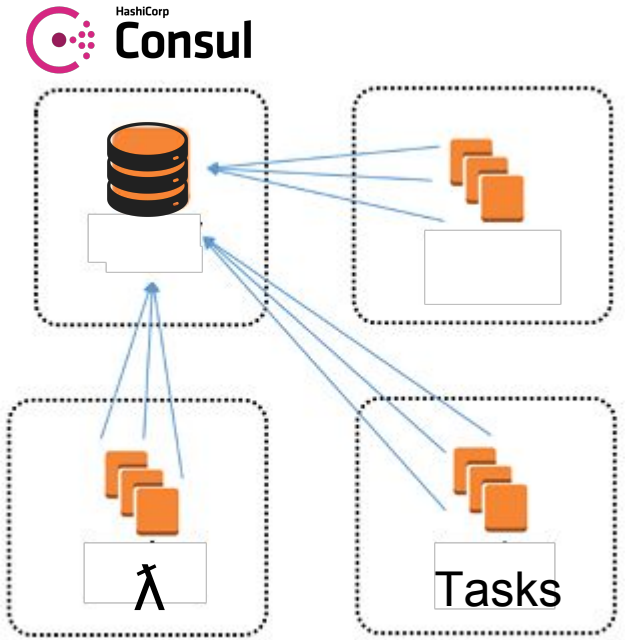
Is your team better than AWS managing critical, highly available, distributed systems?

Going native configuration management

Hashicorp Consul

~~Service discovery~~

Configuration management (KV storage)



Going native configuration management

Requirements:

- Always up
- Low maintenance
- Access control
- Central and accessible
- Encryption for secrets
- Auditable

Going native configuration management

Requirements:

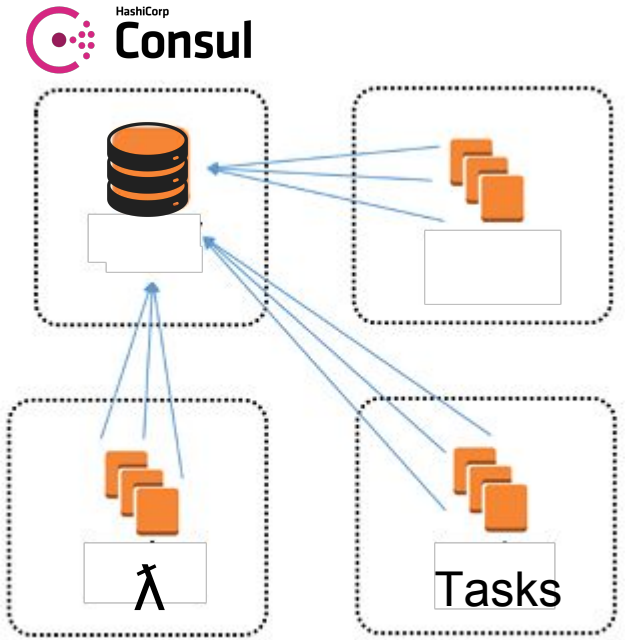
- Always up
- Low maintenance
- Access control
- Central and accessible
- Encryption for secrets
- Auditable

Simplicity makes everyone happy!!

Going native configuration management

Hashicorp Consul

Configuration management (KV storage)

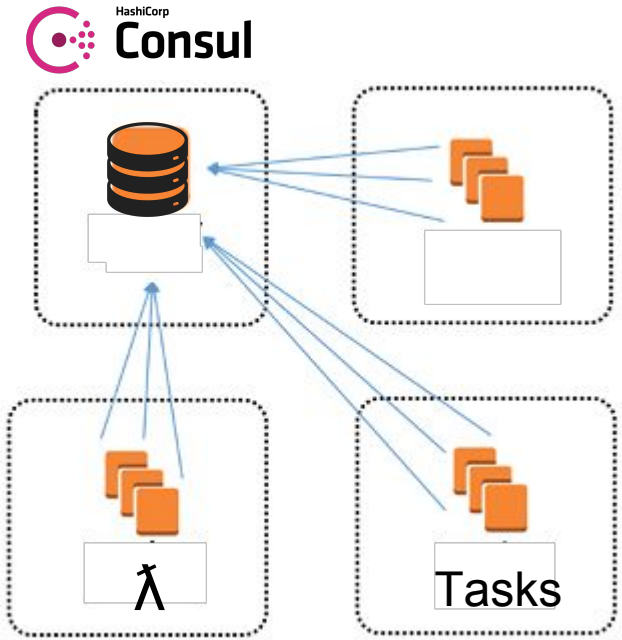


Going native configuration management

Hashicorp Consul

Configuration management (KV storage)

1) services fetch configs at bootstrap
2) services receive notification when configurations change (envconsul)

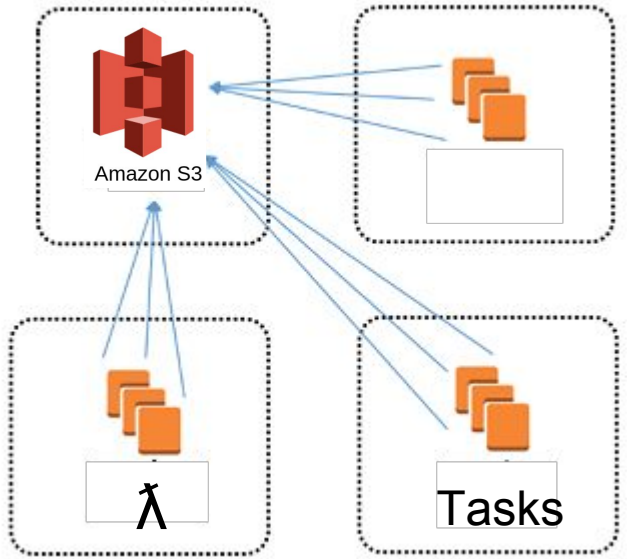


Going native configuration management

Going AWS native (v1)

1) consumer fetch configs at bootstrap

2) consumer receive notification when configurations change (envaws/app logic)



Going native configuration management

Requirements:

- Always up
- Low maintenance
- Access control
- Central and accessible
- Encryption for secrets
- Auditable

Going native configuration management

Requirements:

- ✓ Always up
- ✓ Low maintenance
  - Access control
  - Central and accessible
  - Encryption for secrets
  - Auditable

Going native configuration management

Requirements:

- ✓ Always up
- ✓ Low maintenance
- ✓ Access control
  - Central and accessible
  - Encryption for secrets
  - Auditable

Going native configuration management

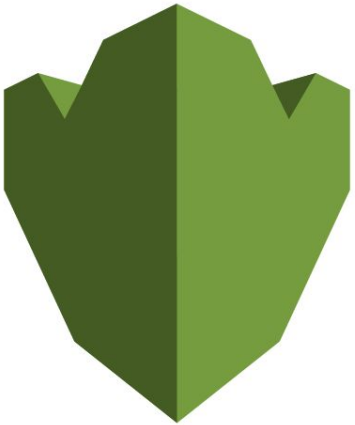
Requirements:

- ✓ Always up
- ✓ Low maintenance
- ✓ Access control
- ✓ Central and accessible
  - Encryption for secrets
  - Auditable

Going native configuration management

Requirements:

- Always up
- Low maintenance
- Access control
- Central and accessible
- Encryption for secrets
- Auditable



Going native configuration management

Requirements:

- Always up
- Low maintenance
- Access control
- Central and accessible
- Encryption for secrets
- Auditable

Going native configuration management

Requirements:

- Always up
- Low maintenance
- Access control
- Central and accessible
- Encryption for secrets
- Auditable
- Simplicity makes everyone happy!!

Going native configuration management

Demo

Going native configuration management

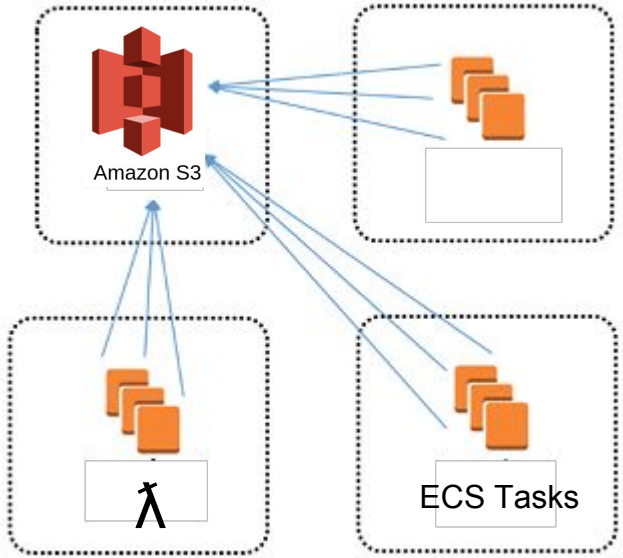
Demo

- ECS services: *foo* and *bar*

1st) Configuration bootstrap

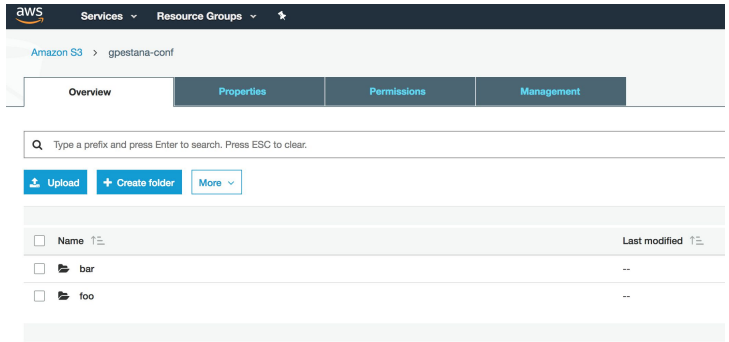
2nd) Tasks killed upon configurations change

- Configuration polling
    - *foo:* in app's logic (node.js module)
    - *bar:* parent process *(envconsul style)*



Going native configuration management

S3 bucket for configurations
- Key per service



IAM role for managing configurations (R/W)

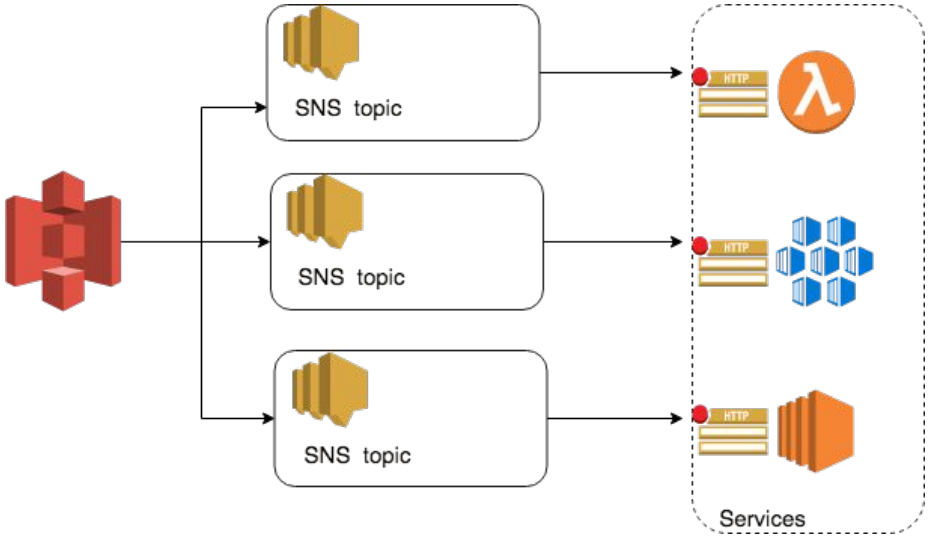
IAM role per consumer for reading service configuration from S3 bucket
- Granular access control

Going native configuration management

Alright cool, but polling is lame!

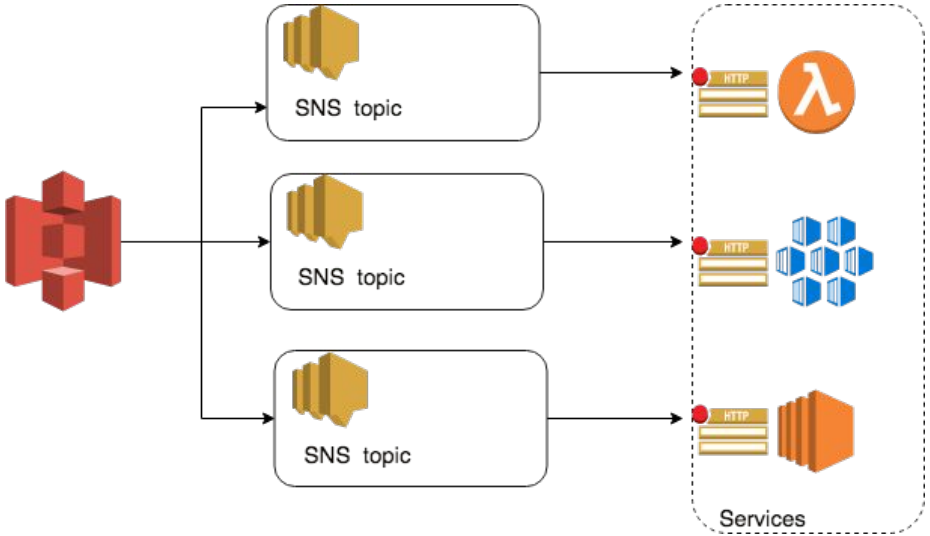
Going native configuration management

Alright cool, but polling is lame!



Going native configuration management

Alright cool, but polling is lame!



`envaws` will provide support to HTTP push notifications

https://github.com/gpestana/envaws

Going native secret management

2 Different models to get configurations updates *(push vs pull)*

A. Service polls for configuration changes

1) Service fetches configuration from S3. Service polls changes in configuration and reboot/hot patch configurations. (ECS, EC2, lambda)

B. Configuration changes are pushed to Service

2) S3 sends notification to SNS, which fans-out notification to correct SQS queues. Services consuming from correct SQS queues receive notifications about new configurations

3) Service exposes endpoint which SNS calls directly upon S3 notification

Going native secret management

Where should the configuration fetching/polling logic live?

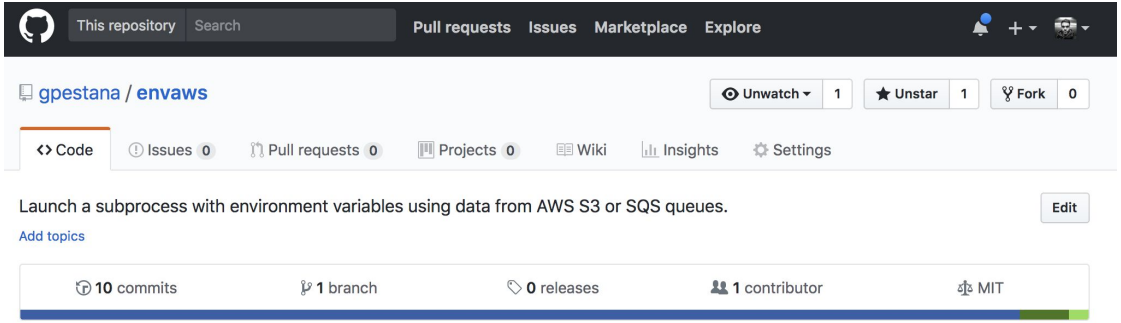
- **Bootstrap process which fetches configurations, populates env environment and launches new sub process with same enc environment**
  - e.g. *envconsul* approach
  - Language agnostic
  - Harder to hot load configurations once they changed
  - Does not work with e.g. lambda functions (??)

- **Library**
  - Not language agnostic
  - Possible to hot load configurations when these change
  - More flexibility
  - Works with lambda functions

Going native secret management

Where should the configuration fetching/polling logic live?

- **Bootstrap process which fetches configurations, populates env environment and launches new sub process with same enc environment**
  - e.g. *envconsul* approach
  - Language agnostic
  - Harder to hot load configurations once they changed
  - Does not work with e.g. lambda functions



Going native secret management

How about secret management?

No time today, but add KMS to the mix!

Wrap up

- It is hard to maintain critical systems in production (e.g. configuration and secret management)

- AWS services can takes us far in config and secret management
  - S3, IAM, KMS, SNS, SQS, audit logs ...

- Many different architectures. Simplest one rely only on S3 and polling by config. consumers

- Configuration polling on application level VS parent-process level

Configuration and secret management in AWS

Code and slide deck

https://github.com/gpestana/aws-conf-management-talk

envaws

https://github.com/gpestana/envaws (use & contribute!)

Discussion

https://twitter.com/gpestana
@gpestana


aws
COMMUNITY DAY
NORDICS
Thanks!
March 21, 2018