Automatic Provisioning of Consul + Vault

whoami?

Misha Manulis

misha@manulis.com

 misham

 @mmanulis

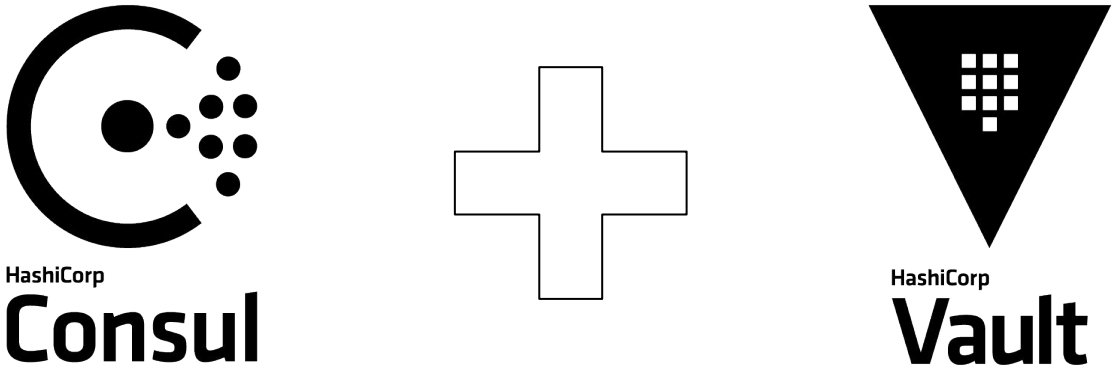
Consultant Extraordinaire at OSG

This programming is brought to you by

- Over 25 years in the business
- Solutions for F1000 clients
- Software dev to IT transformation
- http://www.osgcorp.com

Common Problem

- Configuration data is baked into systems
- Secrets are either baked in or passed around from person to person



What is Consul

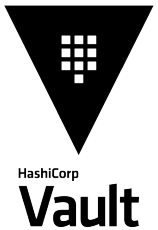
- Service discovery
- Key/Value storage
- Health checks

Why Consul

- Original support for Vault backend
- Advanced capabilities
- Flexible workflows
- Simple deployment

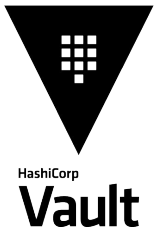
What is Vault

- Tool to manage secrets
- Key rolling
- SSL/TLS cert manager
- Strong ACL



Why Vault

- Open source
- Free with commercial support available
- Specializing in secret management
- Cloud-native
- More than just secret storage
  - Integration with other tools
    - PostgreSQL
    - SSH



How did I end up using Consul & Vault?

- Constantly churning team
  - bench resources, pulled for billable work
  - how many have seen that movie?
- Looking to reduce drag on delivery cadence
  - reduce knowledge transfer
  - easy deploy without a lot of training
- Immutable Infra looked like the answer

Immutable Infrastructure

- What
  - Immutable infrastructure is comprised of immutable components that are replaced for every deployment, rather than being updated in-place. *
- Except
  - Config & secrets are baked into the image
  - No safeguards from running prod images in QA envs

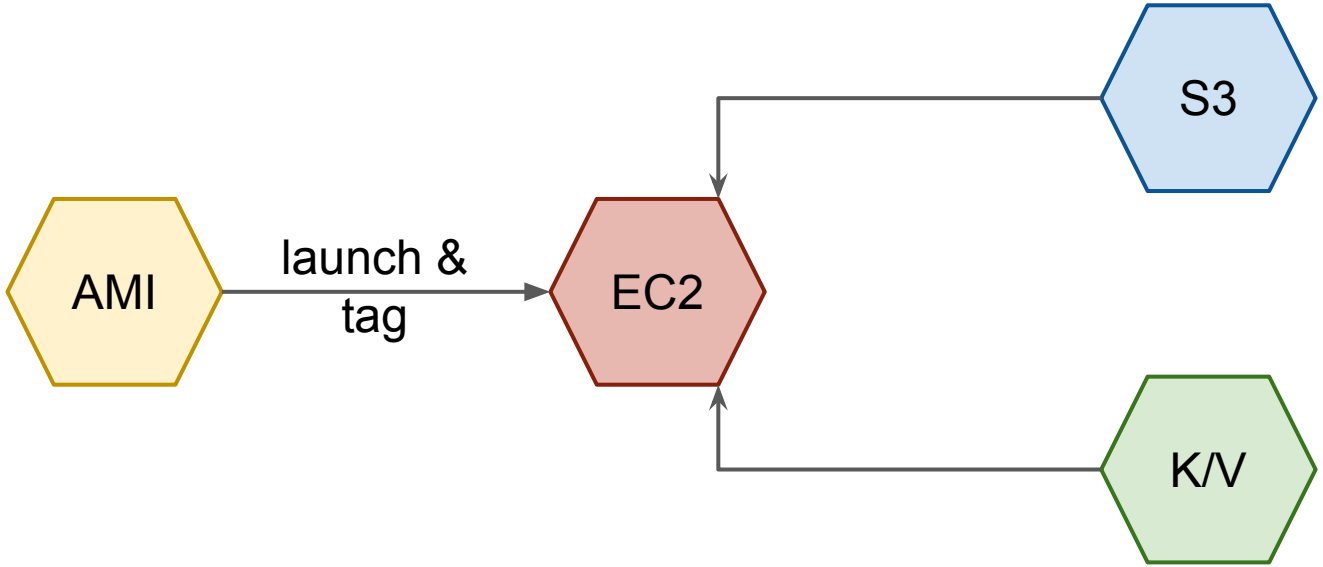
* https://blog.codeship.com/immutable-infrastructure/

My Goals

- No changes to running instances
- All changes are made only via code
- No pre-baked config or secrets
- Automated deploys

The Approach

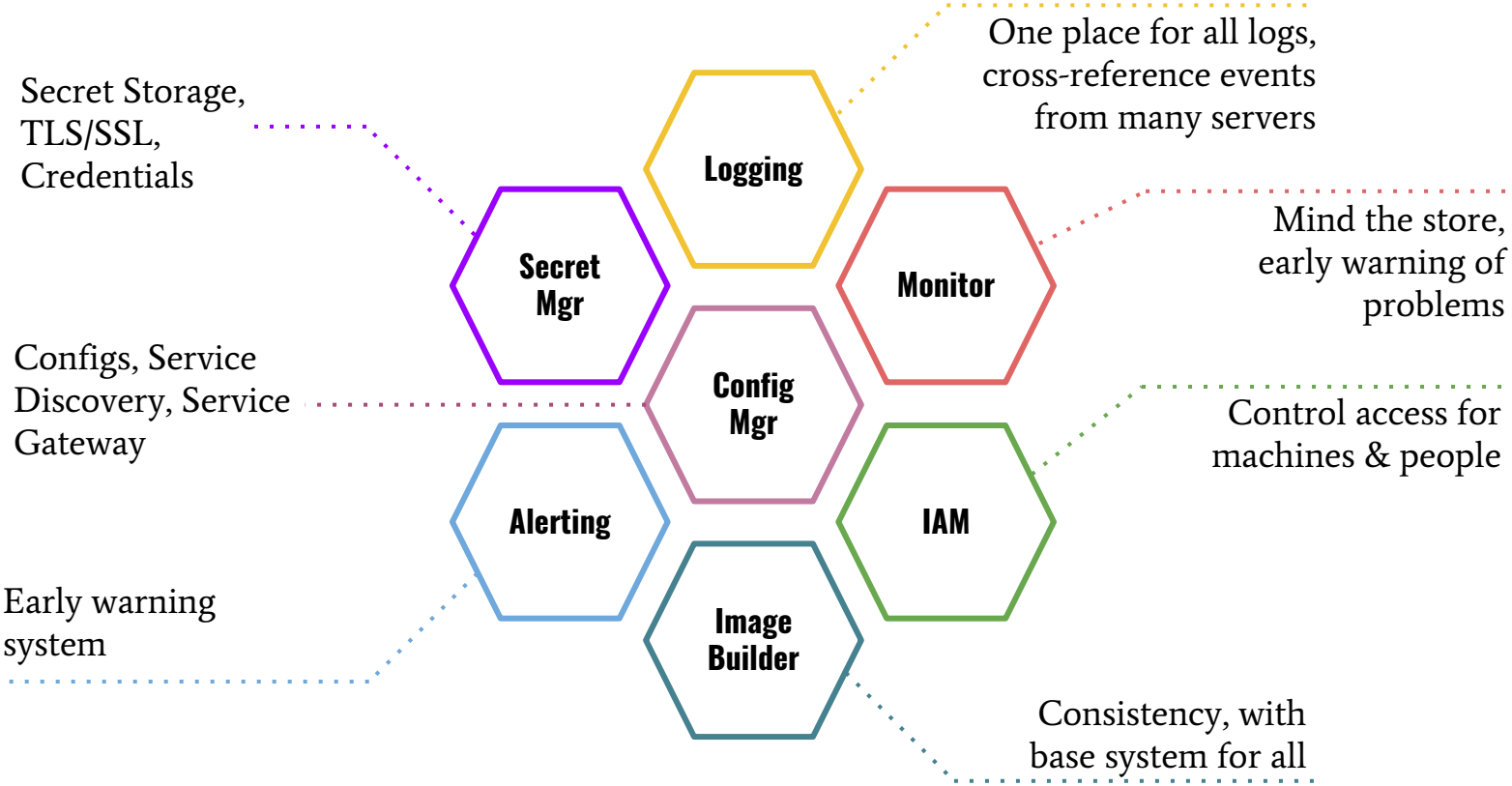
- AMIs + metadata + scripts + configs in S3 / DynamoDB



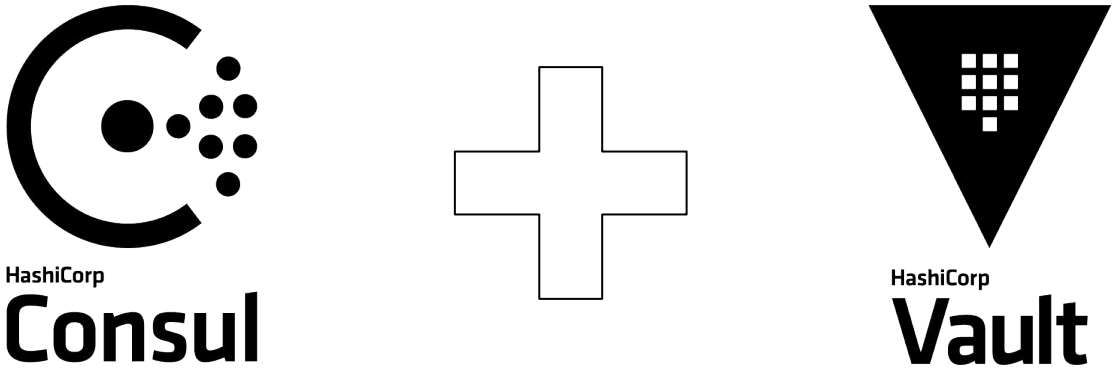
Enterprise World

- 3000+ applications
- 3 continents
- Traditional IT approach
- Cloud migration to keep up with business needs

Cloud Data Center



Our Solution



New Approach

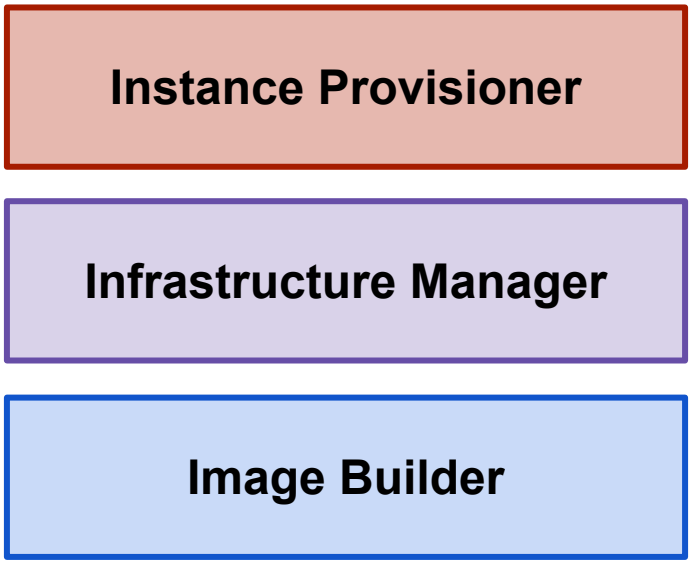


Image Builder - Packer + Ansible

Instance Provisioner

Infrastructure Manager

Image Builder

Image Builder - Ansible

- Why Ansible
- What does Ansible do for us?

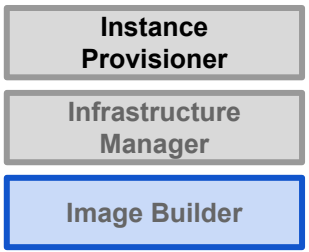
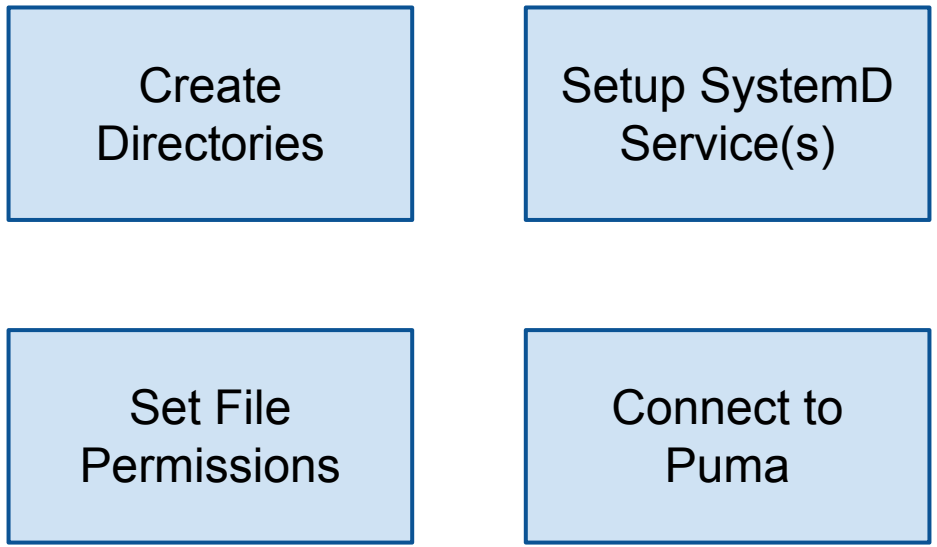


Image Builder - Ansible Example - Rails App



Instance Provisioner

Infrastructure Manager

Image Builder

Image Builder - Packer

Consul Example

AWS Connection Info

Ansible Playbooks

VirtualBox Info

Vagrant Box Creation

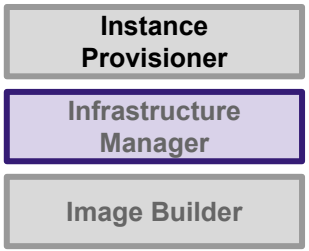
Instance Provisioner

Infrastructure Manager

Image Builder

Infrastructure Manager - Terraform

- What instead of How
- Flexible workflows
- Support for many providers
- Easy to add your own providers & plugins



Terraform Example - Consul

Instance Provisioner

Infrastructure Manager

Image Builder

Instance Provisioner

- Application-specific logic only
- Limit exposure to just the single instance
- Composition over Inheritance
- Single Responsibility Principle

Instance Provisioner

Infrastructure Manager

Image Builder

Instance Provisioner - Implementation

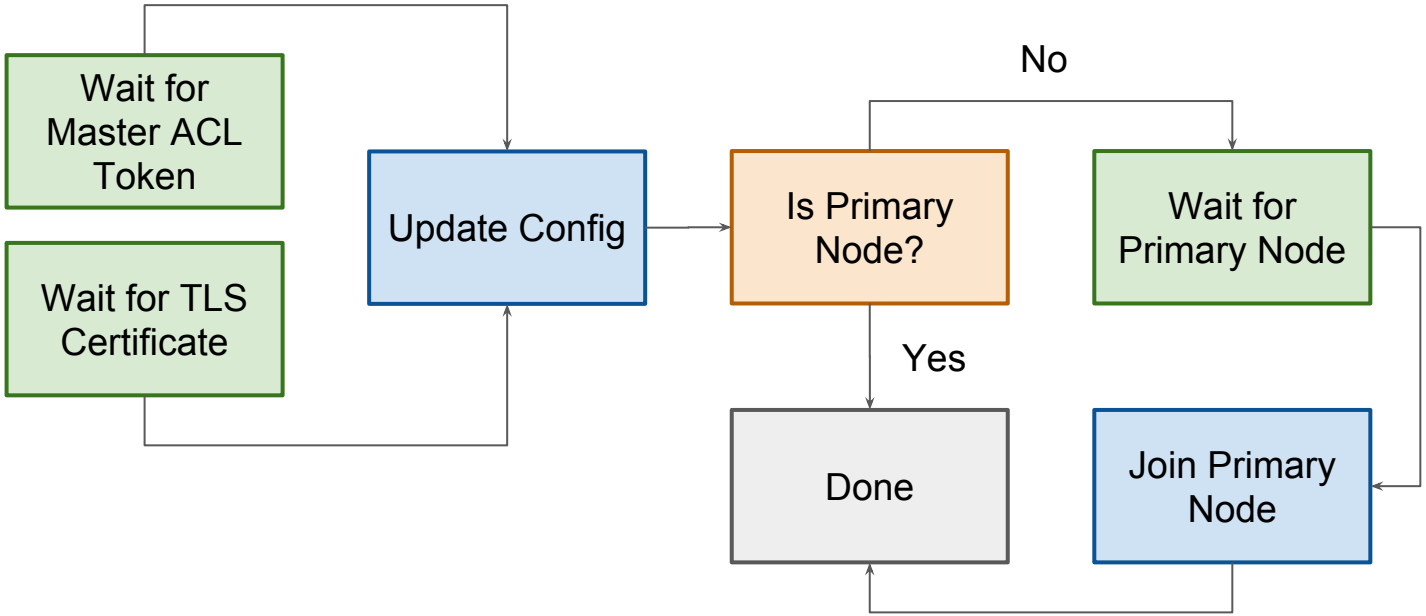
- Bash is not the answer
- Golang
  - It's a distributed system
  - Simple deploys

Instance Provisioner

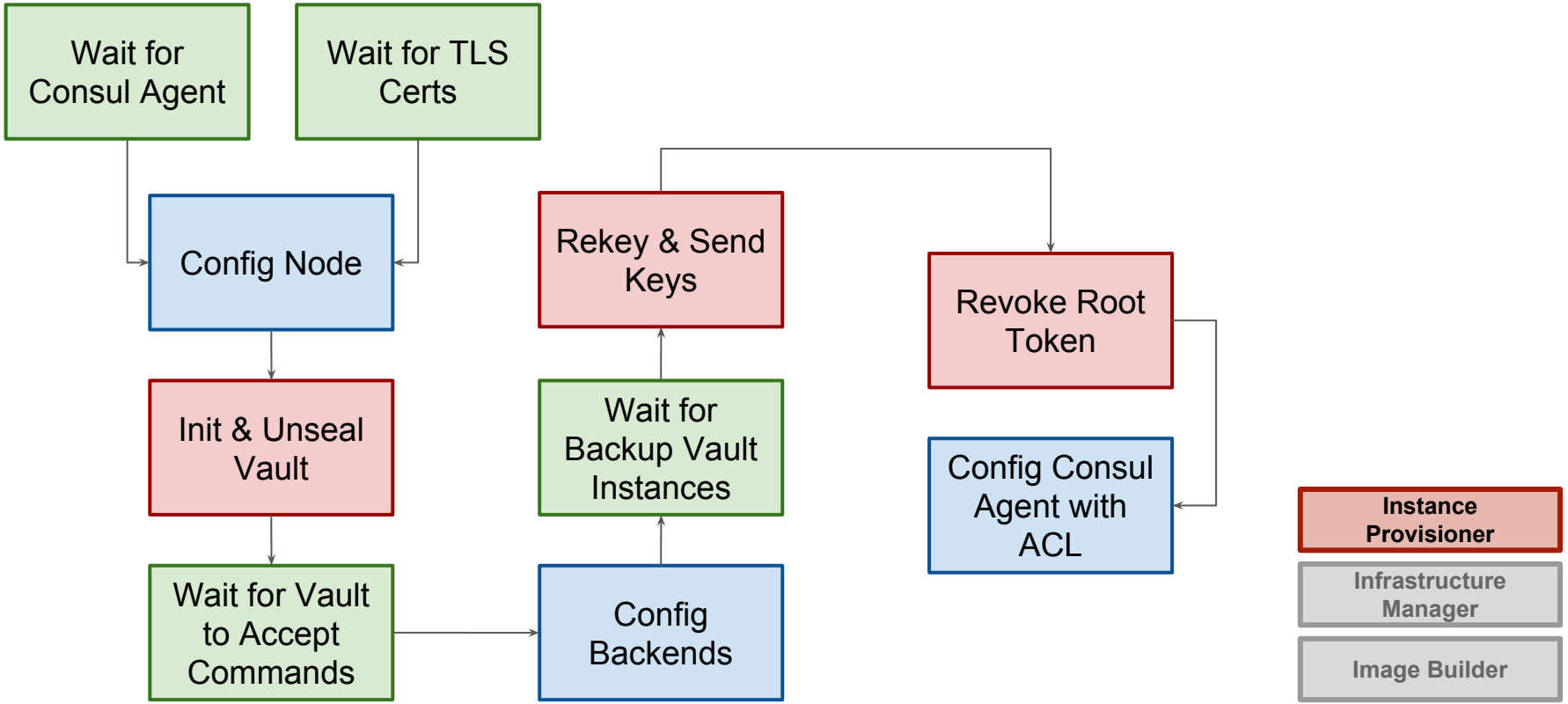
Infrastructure Manager

Image Builder

Bootstrapping Consul Server - Provisioning



Bootstrapping Vault (Primary Node) - Provisioning



Lessons Learned

- Use Vagrant to test your playbooks / recipes before integrating with Packer
- Packer
  - Use existing templates to save time
- Terraform
  - You can't always escape provisioning. For Consul & Vault we copied over TLS certs and ACL tokens
  - Fight to keep it down to absolute minimum
  - Not good for dependency management
- Vault
  - Use PGP, it makes bootstrapping a lot safer
- HashiCorp tools improve very quickly, use CHANGELOG files to stay up to date

Next Steps

- Segment.io Stack (App-centric)
  - https://segment.com/blog/the-segment-aws-stack/
  - https://github.com/segmentio/stack
- Code

Questions?