

Managing secrets in your cloud environment

AWS, GCP, and containers (and beyond)



Hello!

Evan Johnson

Security Eng at Segment



@ejcx_



Segment

Maya Kaczorowski

Security PM at Google



@MayaKaczorowski



Google Cloud Platform



What's a **secret**?

Credentials, configurations, API keys, and other small bits of information needed by applications at build or run time

A close-up photograph of a person's face, focusing on the nose, mouth, and chin. The person has fair skin with freckles and is holding their right index finger vertically against their lips in a 'shh' gesture. The background is dark and out of focus.

“

Secrets... are the
very root of cool

William Gibson, Spook
Country

”

OneLogin security c of data breach

Two breaches in as many years.
company's chief information sec



By Zack Whittaker for Zero Day | June 7, 2017 -- 12:4

4

f

in

t

CEO of Trustico emails 23,000 HTTPS private keys, triggering panicked mass-revocation



White Papers provided by Hewlett-F
Enterprise, Inc.

FROM THE BOING BOING SHOP



News

Uber Discloses Year-Old AWS Data Breach, Exposing Millions of Users

By Gladys Rama ■ 11/21/2017

On Tuesday, ride-sharing app Uber disclosed that its Amazon Web Services (AWS) account was hacked last year, compromising the personal information of 57 million users worldwide, including 600,000 U.S. drivers.

Uber CEO Dara Khosrowshahi, who came into his post just this past August, said [in a statement](#)



18 The Market for Stolen Account Creds

DEC 17

Past stories here have explored the myriad **criminal uses of a hacked co** ways that **your inbox can be spliced and diced** to help cybercrooks ply **value of a hacked company**. Today's post looks at the price of stolen about any e-commerce, bank site or popul fortunes that an enterprising credenti consignment.

Not long ago in Internet time, your typic password-protected Web site would most l several miscreants who routinely leased ac

Viacom Left Sensitive Data And Secret Access Key On Unsecured Amazon Server

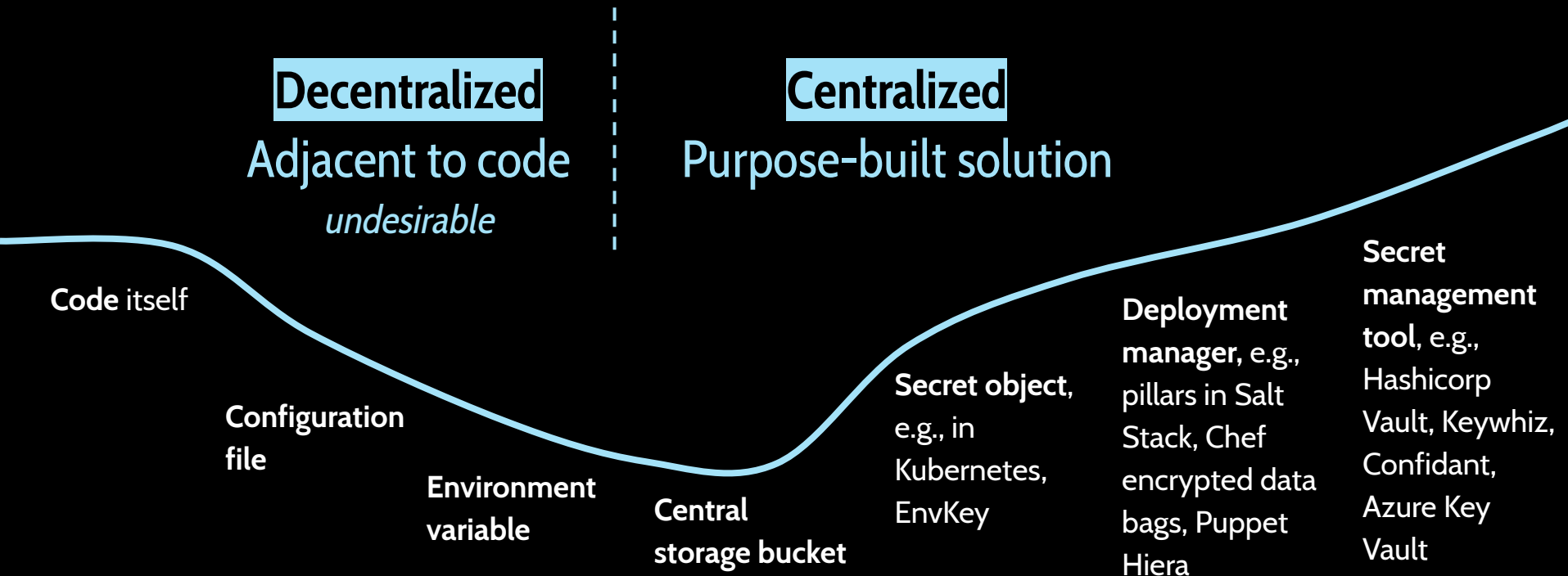
Tuesday, September 19, 2017 Wang Wei



4



How are secrets typically managed?



Wait, people actually keep secrets in code?

KEZZOH



Everybody does it; it's just that nobody talks about it.

Common mistakes made in secret management

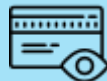
- **Putting secrets in code.** Wait! You just said that! OK, but really don't do it though
- **Not rotating secrets,** or being able to rotate secrets when needed.
- **Not backing up secrets.** Things fail.
- **Not having a concept of identity.** Can't enforce authn/authz without it!
- **Protecting secrets the same way you protect everything else.** They're secret!

Good properties of a secret management solution



Identity

Require strong identities and least privilege



Auditing

Verify the use of individual secrets



Encryption

Always encrypt before writing to disk



Rotation

Change a secret regularly in case of compromise



Isolation

Separate where secrets are used vs managed

Options

Do you run mostly in containers?

No

Standalone

thing

e.g., HashiCorp
Vault

Do you run mostly in one cloud?

No

Yes

Cloud thing

e.g., AWS Secrets
Manager

Yes

Container

thing

e.g., K8s secrets

**‘Cloud
native’ thing**

e.g., K8s + cloud = <3

Options

Do you run mostly in containers?

No

**Standalone
thing**
e.g., HashiCorp
Vault

Yes

**Container
thing**
e.g., K8s secrets

Do you run mostly in one cloud?

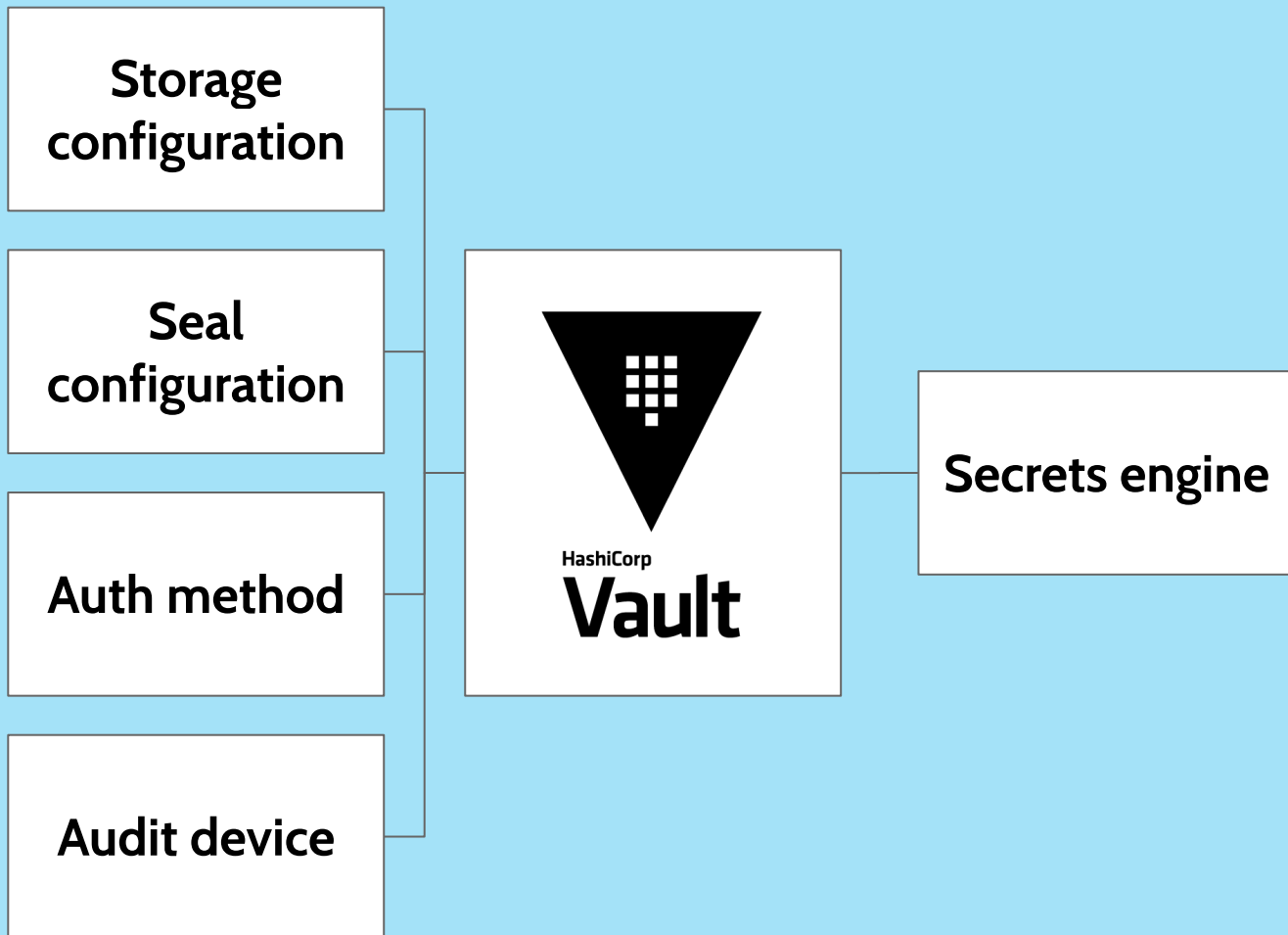
No

Yes

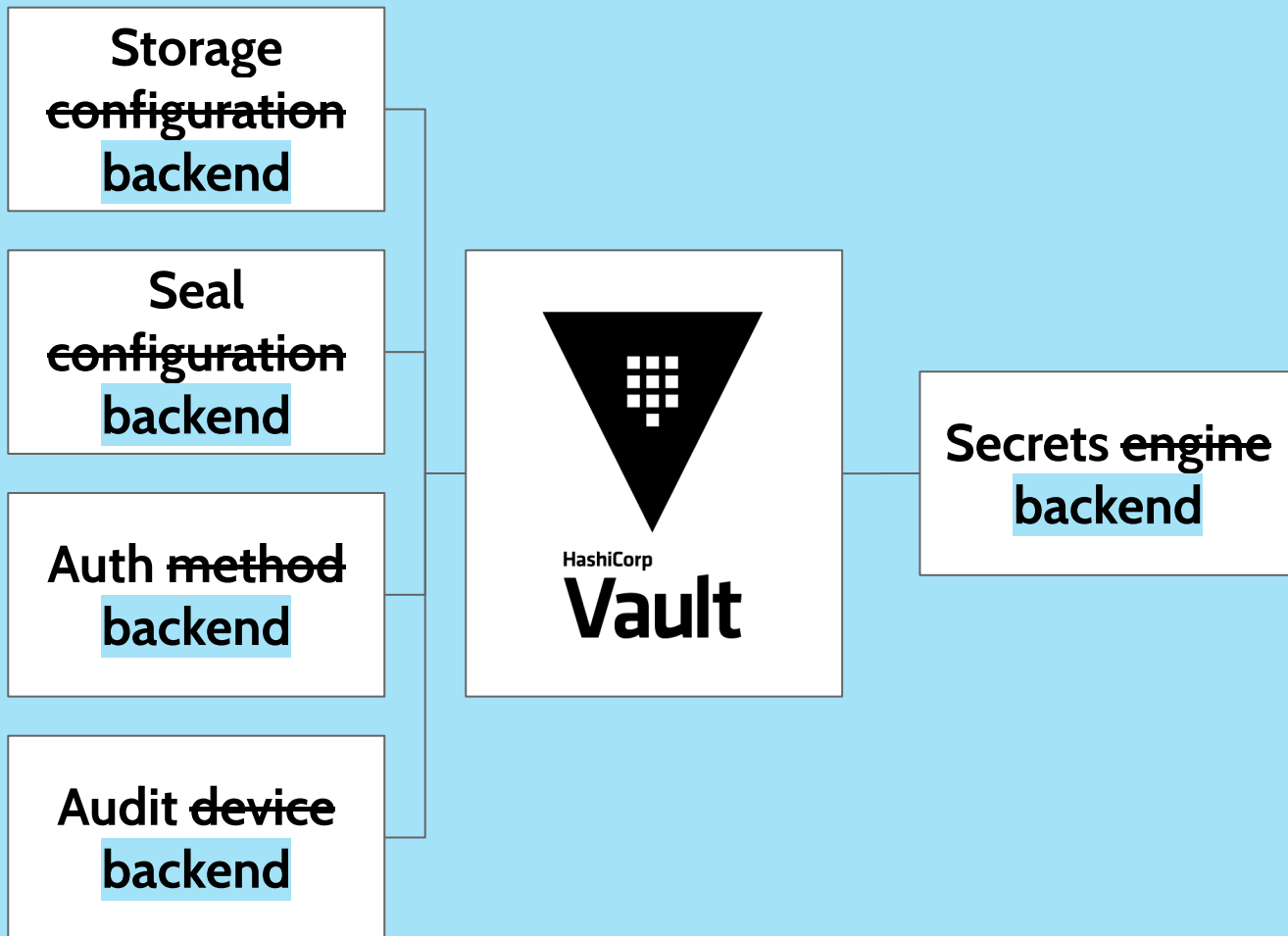
Cloud thing
e.g., AWS Secrets
Manager

**‘Cloud
native’ thing**
e.g., K8s + cloud = <3

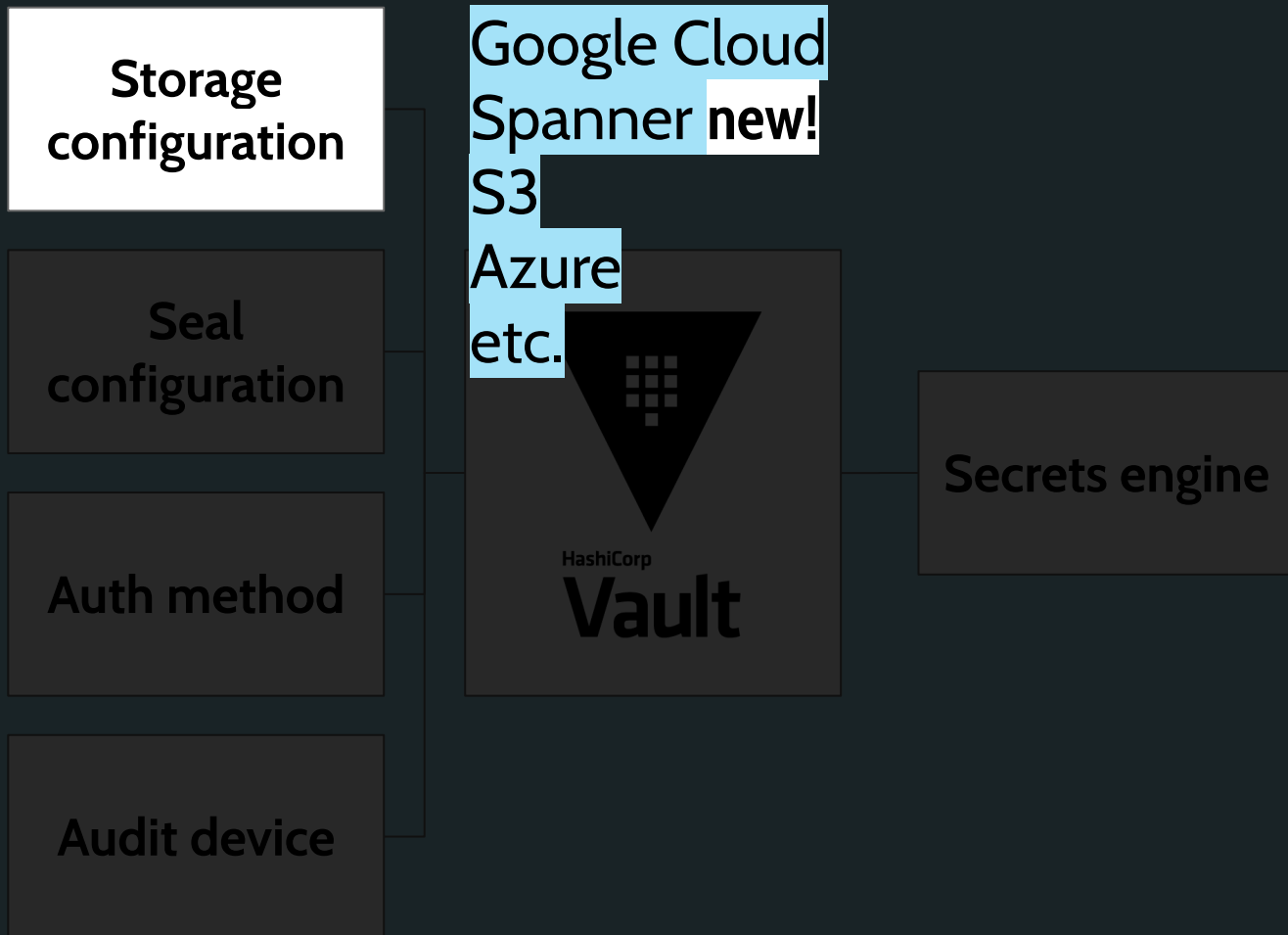
HashiCorp Vault



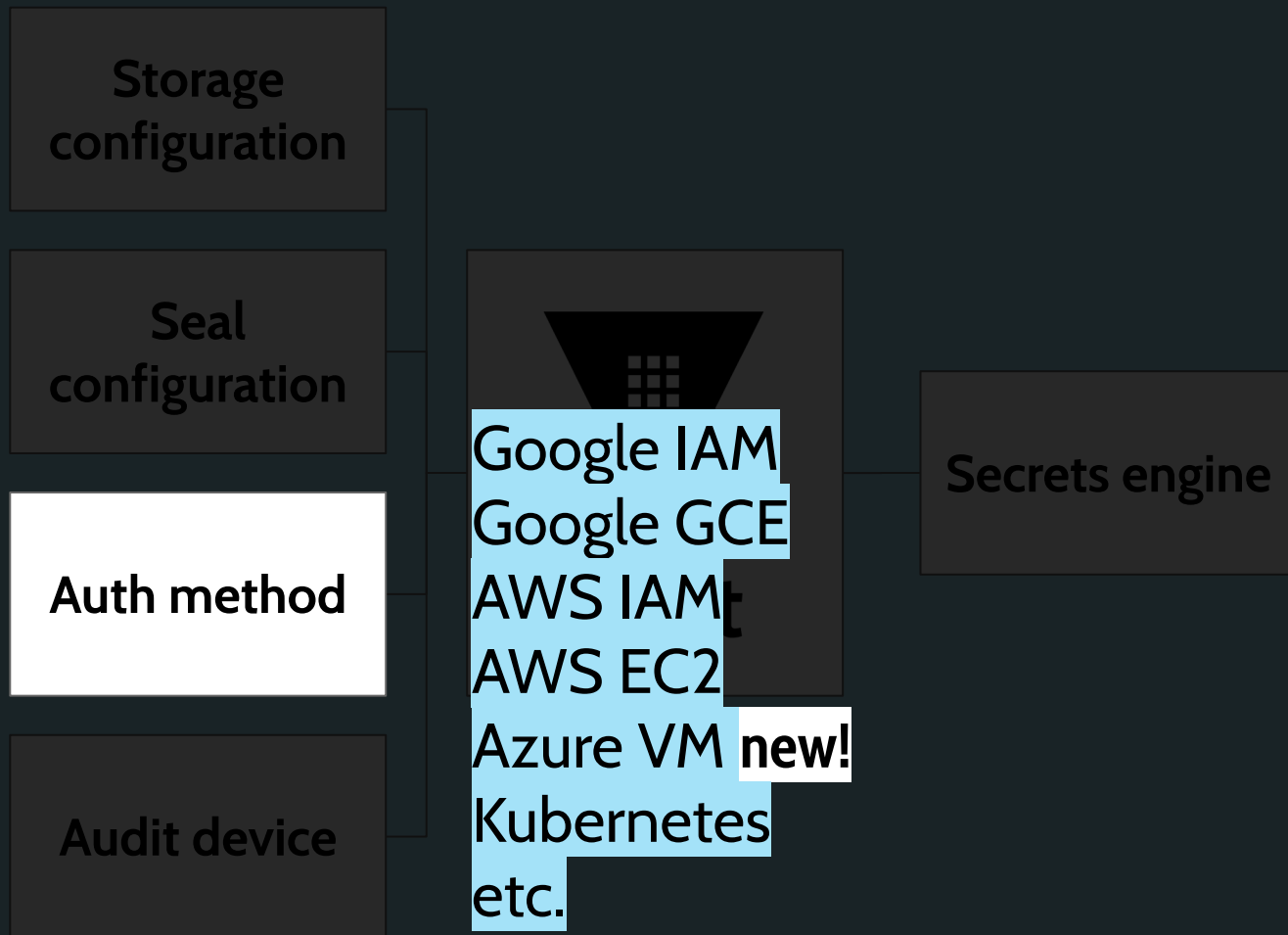
HashiCorp Vault



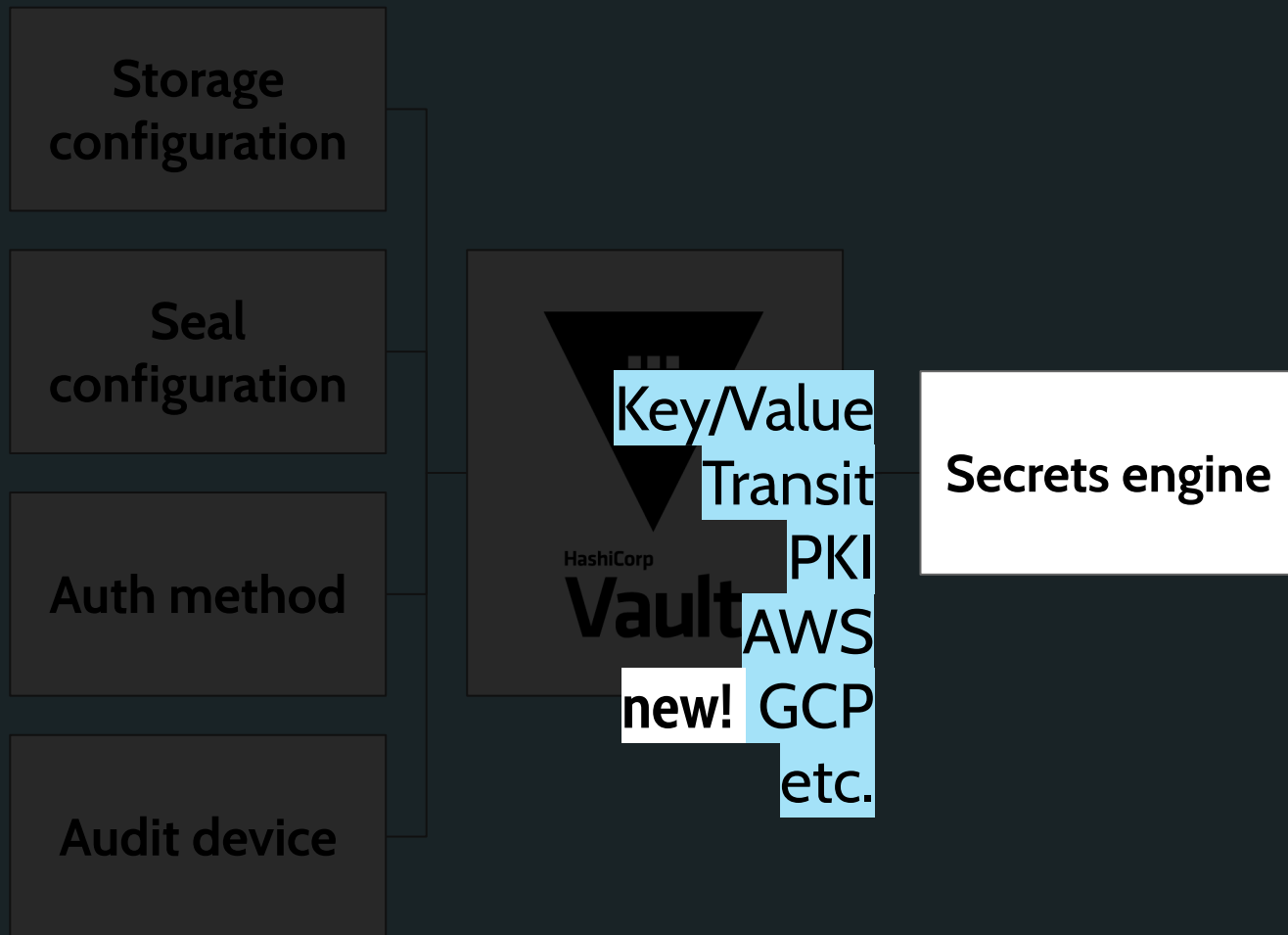
HashiCorp Vault



HashiCorp Vault



HashiCorp Vault



When should you use Vault?

I'm already using Vault on prem

I'm using multiple clouds

I have dedicated engineers

Options

Do you run mostly in containers?

No

**Standalone
thing**

e.g., HashiCorp
Vault

Yes

**Container
thing**

e.g., K8s secrets

Do you run mostly in one cloud?

No

Yes

Cloud thing

e.g., AWS Secrets
Manager

**‘Cloud
native’ thing**

e.g., K8s + cloud = <3

AWS

AWS KMS

Parameter Store

new! Secrets
Manager

Step 1 **Secret type**

Step 2 Name and description

Step 3 Configure rotation

Step 4 Review

[AWS Secrets Manager](#) > [Secrets](#) > Store a new secret

Store a new secret

Select secret type [Info](#)

☒ Credentials for RDS database

☐ Credentials for other database

☐ Other type of secrets (e.g. API key)

Specify the user name and password to be stored for this secret. [Info](#)


User name:

Password:

☐ Show password

Select the encryption key [Info](#)

Select the AWS KMS key to use to encrypt your secret information. You can encrypt using the default service encryption key that AWS Secrets Manager creates on your behalf or a customer master key (CMK) that you have stored in AWS KMS.

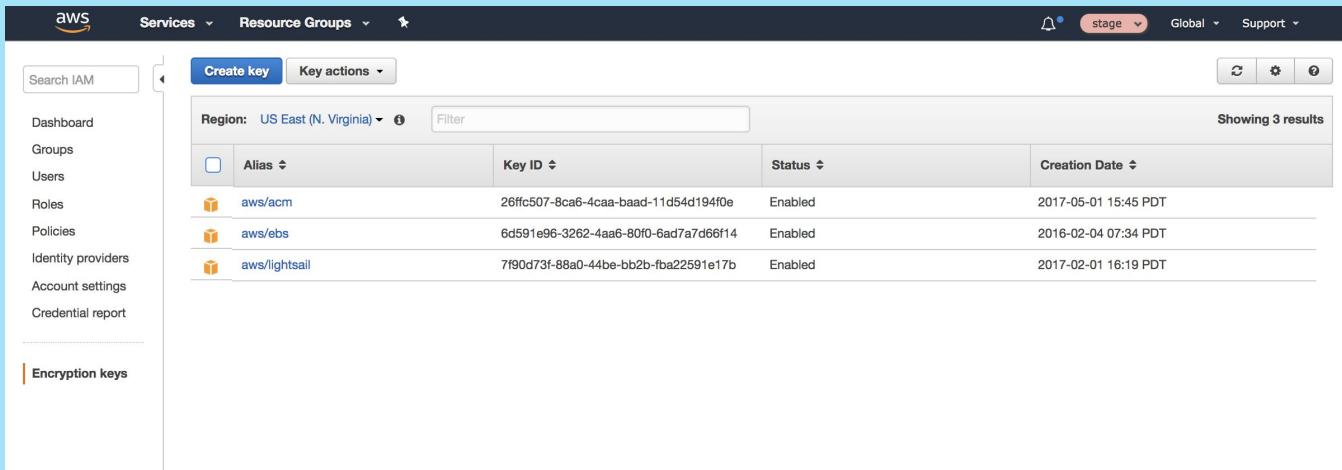
DefaultEncryptionKey ▼ 

[Add new key](#) [↗](#)

<https://aws.amazon.com/blogs/mt/the-right-way-to-store-secrets-using-parameter-store/>

<https://aws.amazon.com/secrets-manager/>

AWS KMS



Search IAM

Create key Key actions

Region: US East (N. Virginia) Filter Showing 3 results

<input type="checkbox"/>	Alias	Key ID	Status	Creation Date
	aws/acm	26ffc507-8ca6-4caa-baad-11d54d194f0e	Enabled	2017-05-01 15:45 PDT
	aws/ebs	6d591e96-3262-4aa6-80f0-6ad7a7d66f14	Enabled	2016-02-04 07:34 PDT
	aws/lightsail	7190d73f-88a0-44be-bb2b-fba22591e17b	Enabled	2017-02-01 16:19 PDT

Encrypt

Decrypt

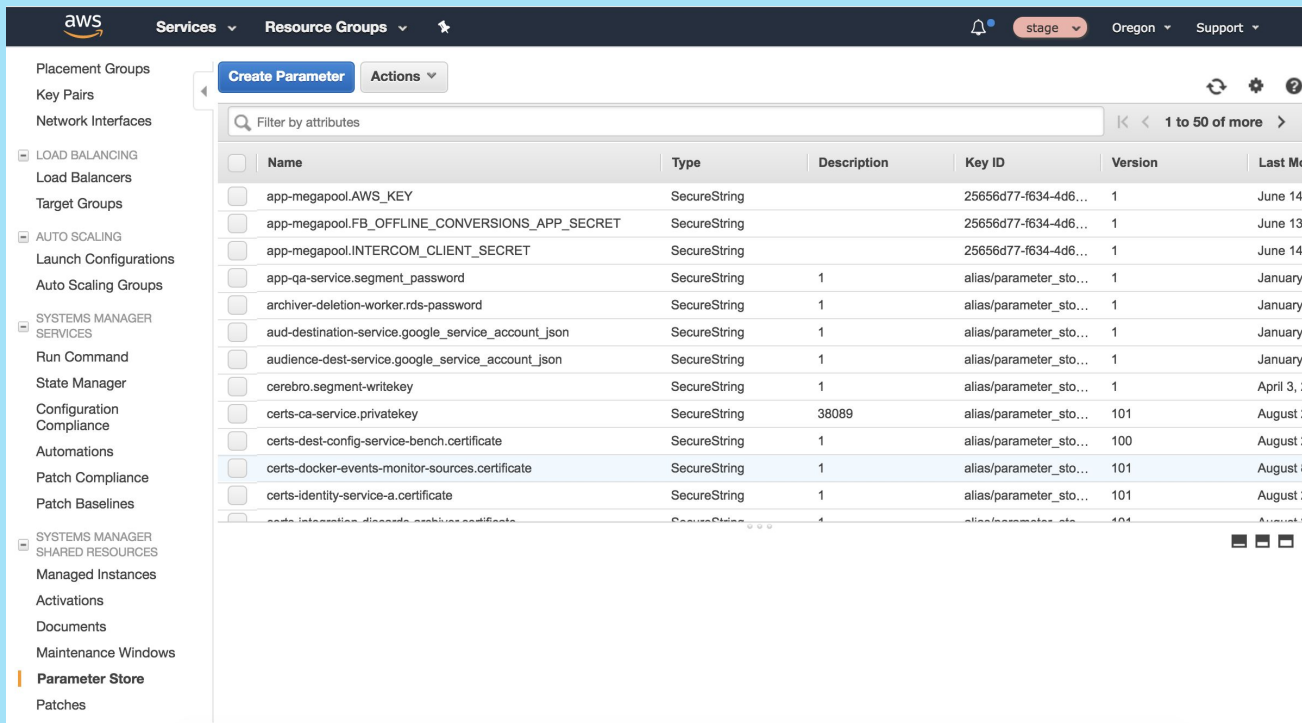
**Generate
Random Data**

**Grant 3rd Party
Access**

**Automatic Key
Rotation**

**Customer or
AWS managed**

AWS SSM Parameter Store



The screenshot shows the AWS SSM Parameter Store console. The left sidebar contains a navigation menu with categories like Placement Groups, Key Pairs, Network Interfaces, LOAD BALANCING, AUTO SCALING, SYSTEMS MANAGER SERVICES, and SHARED RESOURCES. The 'Parameter Store' option is highlighted. The main area shows a table of parameters with columns for Name, Type, Description, Key ID, Version, and Last Modified. The table lists various parameters, including app-megapool.AWS_KEY, app-megapool.FB_OFFLINE_CONVERSIONS_APP_SECRET, and app-megapool.INTERCOM_CLIENT_SECRET. The 'Create Parameter' button is visible at the top left of the main area.

Name	Type	Description	Key ID	Version	Last Modified
app-megapool.AWS_KEY	SecureString		25656d77-f634-4d6...	1	June 14
app-megapool.FB_OFFLINE_CONVERSIONS_APP_SECRET	SecureString		25656d77-f634-4d6...	1	June 13
app-megapool.INTERCOM_CLIENT_SECRET	SecureString		25656d77-f634-4d6...	1	June 14
app-qa-service.segment_password	SecureString	1	alias/parameter_sto...	1	January
archiver-deletion-worker.rds-password	SecureString	1	alias/parameter_sto...	1	January
aud-destination-service.google_service_account_json	SecureString	1	alias/parameter_sto...	1	January
audience-dest-service.google_service_account_json	SecureString	1	alias/parameter_sto...	1	January
cerebro.segment-writekey	SecureString	1	alias/parameter_sto...	1	April 3, 2024
certs-ca-service.privatekey	SecureString	38089	alias/parameter_sto...	101	August 1, 2024
certs-dest-config-service-bench.certificate	SecureString	1	alias/parameter_sto...	100	August 1, 2024
certs-docker-events-monitor-sources.certificate	SecureString	1	alias/parameter_sto...	101	August 1, 2024
certs-identity-service-a.certificate	SecureString	1	alias/parameter_sto...	101	August 1, 2024
certs-identity-service-b.certificate	SecureString	1	alias/parameter_sto...	101	August 1, 2024

Save
“SecureStrings”

Control Access
using IAM

Completely
Managed

AWS SSM Parameter Store

```
vim modules/service/iam.tf
24
25 data "aws_iam_policy_document" "parameter_store_role_policy" {
26   statement {
27     actions = [
28       "ssm:GetParameters",
29     ]
30
31     resources = [
32       "arn:aws:ssm:*:*:parameter/${coalesce(var.secret_label, var.name)}/*"
33     ]
34   }
35
36   statement {
37     actions = [
38       "ssm:DescribeParameters",
39     ]
40
41     resources = [
42       "arn:aws:ssm:*:*:*"
43     ]
44   }
45
```

Secrets stored in Parameter store with the name:
\$service/\$secret

ejcx@Evans-MacBook-Pro: ~

→ ~ aws-okta exec stage -- chamber list gateway-api

Key	Version	LastModified	User
hydra_client_secret	1	07-07 16:13:42	arn:aws:sts::355207333203:assumed-role/ops-admin/1499469222102728881
hydra_pg_password	2	08-11 11:36:20	arn:aws:sts::355207333203:assumed-role/ops-admin/1502476579788251012
mode_access_secret	1	09-25 10:57:57	arn:aws:sts::355207333203:assumed-role/ops-admin/1506362276156780275
auth0_client_secret	1	08-31 16:52:11	arn:aws:sts::355207333203:assumed-role/ops-admin/1504223530549137036
mode_secret_key	1	09-25 10:59:19	arn:aws:sts::355207333203:assumed-role/ops-admin/1506362358425939592
b64_bigquery_private_key	1	11-06 06:41:55	arn:aws:sts::355207333203:assumed-role/ops-admin/1509979317771724450
engine_api_key	2	10-24 17:00:43	arn:aws:sts::355207333203:assumed-role/ops-admin/1508889638748088000
new_relic_key	2	08-14 17:22:43	arn:aws:sts::355207333203:assumed-role/ops-admin/1502756556181958581
auth0_secret	1	08-31 10:18:28	arn:aws:sts::355207333203:assumed-role/ops-admin/1504199907677841864

→ ~

<https://github.com/segmentio/chamber>

AWS SSM Parameter Store

The screenshot displays the AWS Management Console interface. On the left-hand navigation pane, the 'Parameter Store' link is highlighted with a red rectangular box. The main content area shows the 'Resources' section for the US West (Oregon) region, listing various EC2 resources such as Running Instances, Elastic IPs, Snapshots, Volumes, Load Balancers, Key Pairs, and Security Groups. Other sections visible include 'Create Instance', 'Service Health', 'Scheduled Events', 'Account Attributes', and 'AWS Marketplace'.

Navigation Menu (Left):

- Placement Groups
- Key Pairs
- Network Interfaces
- LOAD BALANCING
 - Load Balancers
 - Target Groups
- AUTO SCALING
 - Launch Configurations
 - Auto Scaling Groups
- SYSTEMS MANAGER SERVICES
 - Run Command
 - State Manager
 - Configuration Compliance
 - Automations
 - Patch Compliance
 - Patch Baselines
- SYSTEMS MANAGER SHARED RESOURCES
 - Managed Instances
 - Activations
 - Documents
 - Parameter Store**

Resources (Main Content):

You are using the following Amazon EC2 resources in the US West (Oregon) region:

Resource Type	Count
Running Instances	235
Elastic IPs	4
Dedicated Hosts	0
Snapshots	7173
Volumes	951
Load Balancers	233
Key Pairs	5
Security Groups	209
Placement Groups	0

Service Health:

Service Status: US West (Oregon): This service is operating normally.

Availability Zone Status:

- us-west-2a: Availability zone is operating normally
- us-west-2b: Availability zone is operating normally

Scheduled Events: US West (Oregon): 1 instances have scheduled events.

Account Attributes:

- Supported Platforms: VPC
- Default VPC: none
- Resource ID length management

Additional Information:

- Getting Started Guide
- Documentation
- All EC2 Resources
- Forums
- Pricing
- Contact Us

AWS Marketplace:

Find free software trial products in the AWS Marketplace from the [EC2 Launch Wizard](#). Or try these popular AMIs:

- Barracuda CloudGen Firewall for AWS - PAYG

Provided by Barracuda Networks, Inc.
Rating ★★★★★
Starting from \$0.60/hr or from \$4,599/yr (12% savings) for software + AWS usage fees
[View all Infrastructure Software](#)

Footer:

Feedback English (US) © 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

What should you do on AWS?

If you have significant % of infra not on AWS, use HashiCorp Vault

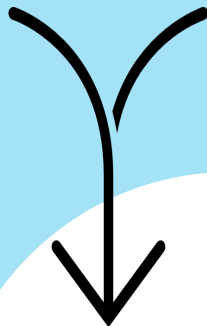
If you don't need rotation, use Chamber and Parameter Store

If you make lots of requests, build something with AWS KMS and S3

If you need rotation or special requirements use Secrets Manager

GCP

Cloud KMS



Protected with



Cloud KMS



IAM



Cloud Audit Logging

<https://cloud.google.com/kms/docs/store-secrets>

What should you do on GCP?

If you need general security and redundancy, use **Cloud Spanner** or **GCS**

If you just need encryption, encrypt your secrets manually using **Cloud KMS** and store them in **Cloud Spanner**

If you need rotation, versioning, or more complex functionality, use **HashiCorp Vault**

Options

Do you run mostly in containers?

No

**Standalone
thing**
e.g., HashiCorp
Vault

Do you run mostly in one cloud?

No

Yes

Cloud thing
e.g., AWS Secrets
Manager

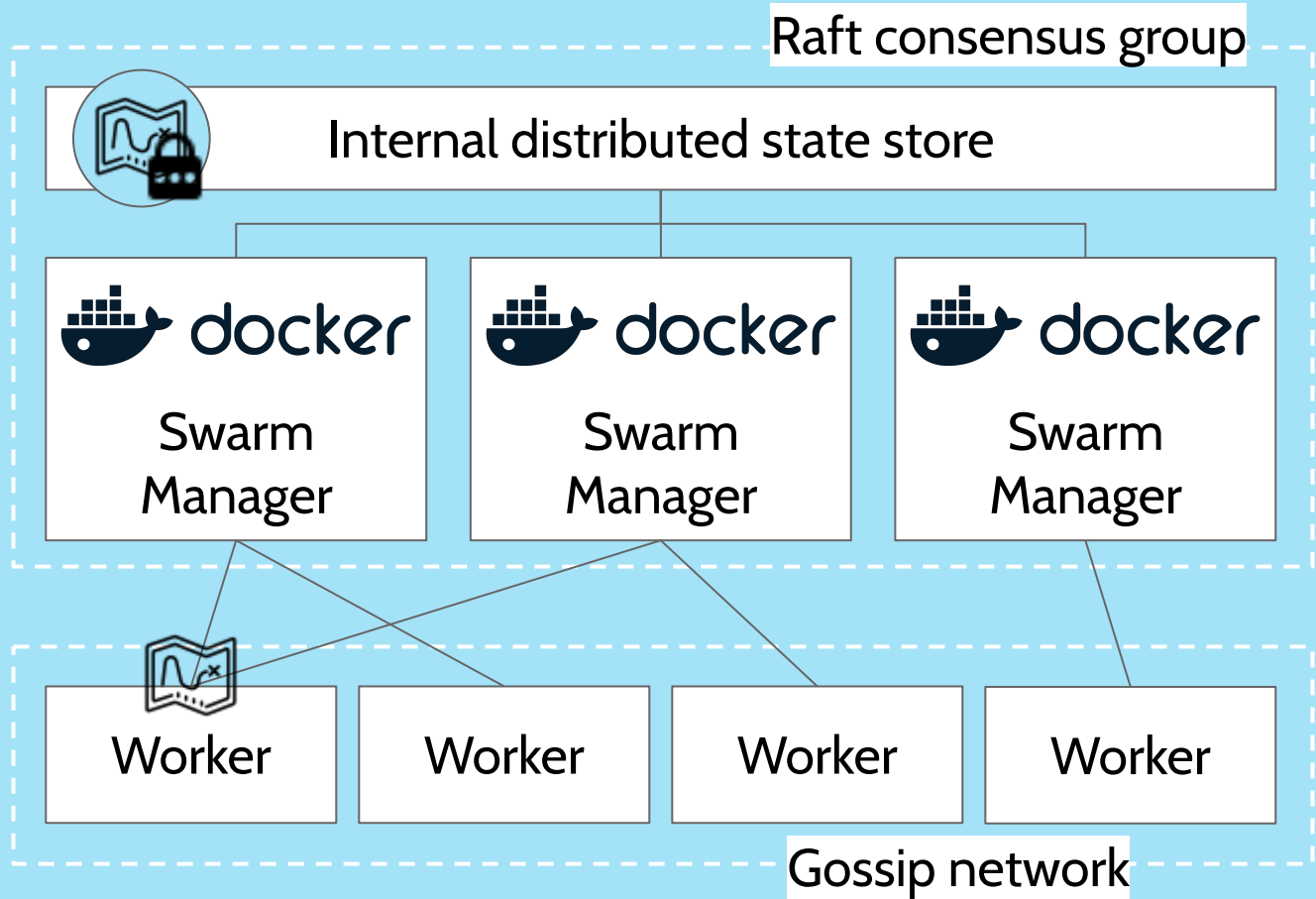
Yes

**Container
thing**
e.g., K8s secrets

**‘Cloud
native’ thing**
e.g., K8s + cloud = <3

Docker Swarm

Secrets



<https://docs.docker.com/engine/swarm/secrets/>



Kubernetes

Secrets

Kubernetes 1.7

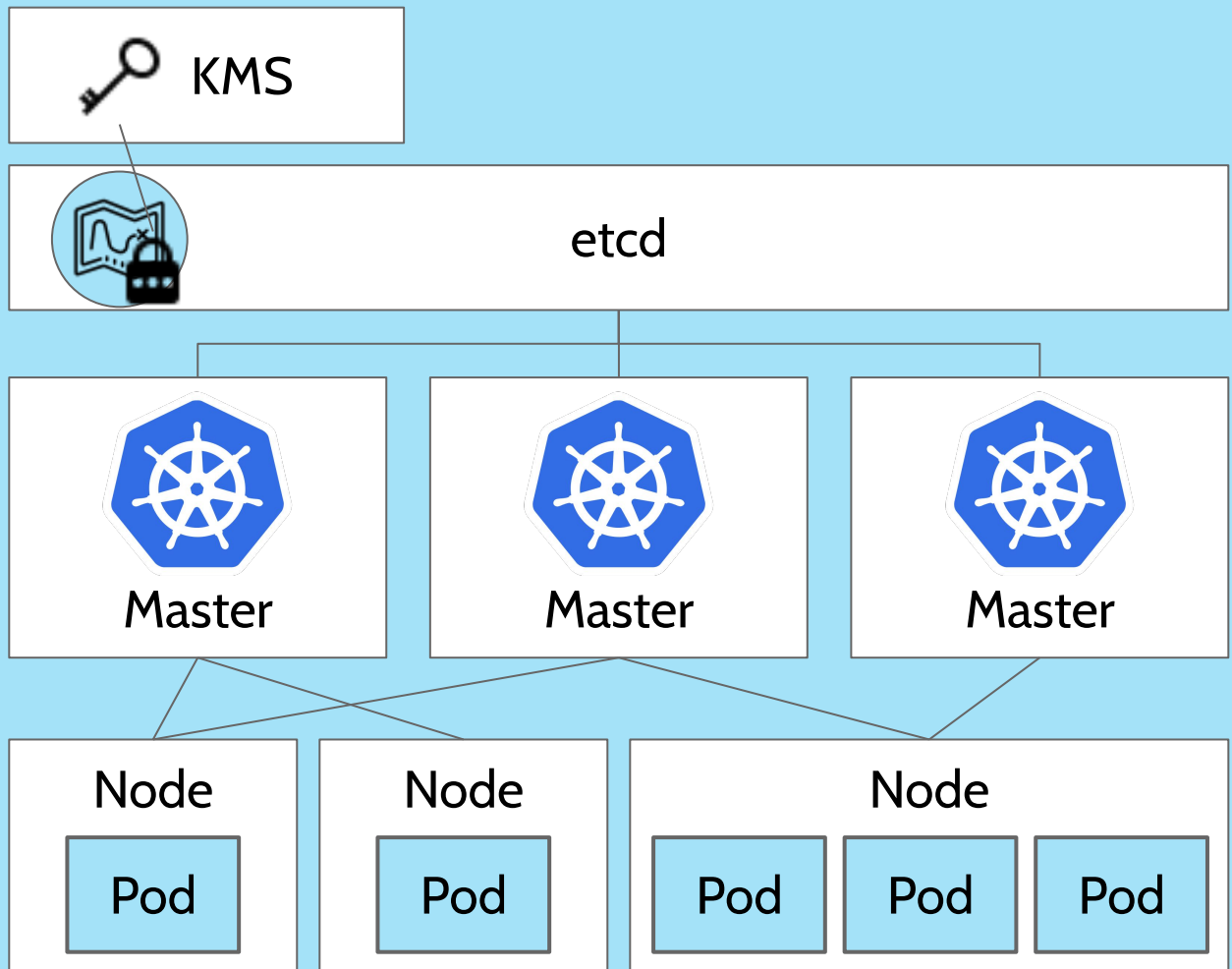
Secrets encryption

Kubernetes 1.10

KMS encryption

Azure Key Vault

Google Cloud KMS



Summary

Standalone

HashiCorp Vault

Cloud

AWS Parameter Store

AWS Secrets Manager

GCP Cloud KMS

Container

Docker Swarm

Kubernetes



Identity



Auditing



Encryption



Rotation



Isolation

All solutions have some concept of identity, auditing, and isolation

→ Encryption and rotation are differentiators, so look for that if you can

Almost any solution meets basic security needs

→ You'll likely choose a solution based on your environment and usability

Residual risks

What's still hard?

- **Usability.** It's great to have these tools, now figure them out without messing up.
- **Root secret.** How do you protect the secret to all secrets?
- **Secret rotation.** Some tools do it, some don't; but it's still highly manual in most cases.

Secret management is hard

Don't keep secrets in source code

Encrypt everything



Thanks!

Questions?