



SNIA[®] STORAGE
SECURITY SUMMIT
Wednesday, May 11, 2022 • Virtual

Let Your Object Storage Save You From the Bad Guys!

Yuval Lifshitz, Principal Software Engineer, Red Hat



A SNIA[®] Event

Agenda

- Little bit on ransomware
- Some Information Theory background
- Ceph Storage System with Object Store
- How Rook makes everything easy
- Lua on the Object Store Gateway
- Tie it all together

Ransomware - What we (think that we) know

Ransomware is here to stay

The incentive for the perpetrators is too strong for them to give up on such a powerful tool

The first line of defence will always break

Via social engineering, zero-day attack or other means

Detecting ransomware while it is incubating is hard

This is the critical time after infection and before the ransomware make itself known.

During that time we still have a chance to save our data, so the ransomware does it best to hide

Storage Behavioral Analysis

- Behavior of the ransomware to detect its existence during incubations
- Behaviors that are useful for the ransomware so it does not try to avoid it
- Behaviors that could be identified as anomalies compared to its operation before the infection
- Behaviors that could be identified outside of the infected clients - ideally at the storage backend which could be considered more secure (smaller attack surface)

Information Theory Entropy

Entropy (from Wikipedia):

“The entropy of a random variable is the average level of “information”, “surprise”, or “uncertainty” inherent to the variable’s possible outcomes.”

Information I of an events E is defined as: $I(E) = -\log_2(p(E))$

So entropy H of a random variable X is:

$$H(X) = -\sum_{i=1}^n P(x_i) \log P(x_i)$$

Note that we use log at the base of 2 for “bit entropy”, meaning that the event being sampled from the distributions is zero or one, and the entropy is measured in “bits of information”

Example of Entropy

Let's look at random variables where the events are letters being sampled from an alphabet of the 26 letters and a space. In this case this is not “bit entropy”, and we would use “normalized entropy” (a.k.a “efficiency”) as:

$$\frac{H}{H_{max}} = - \sum_{i=1}^n \frac{p(x_i) \log_b(p(x_i))}{\log_b(n)}$$

So the text “*hello world*” gives us following distribution:

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	
0.00	0.00	0.00	0.09	0.09	0.00	0.00	0.09	0.00	0.00	0.00	0.27	0.00	0.00	0.18	0.00	0.00	0.09	0.00	0.00	0.00	0.00	0.09	0.00	0.00	0.00	0.09

And normalized entropy of: 0.598

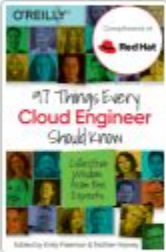





And the text “*the quick brown fox jumps over the lazy dog*” gives us following distribution:

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	
0.02	0.02	0.02	0.02	0.07	0.02	0.02	0.05	0.02	0.02	0.02	0.02	0.02	0.02	0.09	0.02	0.02	0.05	0.02	0.05	0.05	0.02	0.02	0.02	0.02	0.02	0.19

And normalized entropy of: 0.922

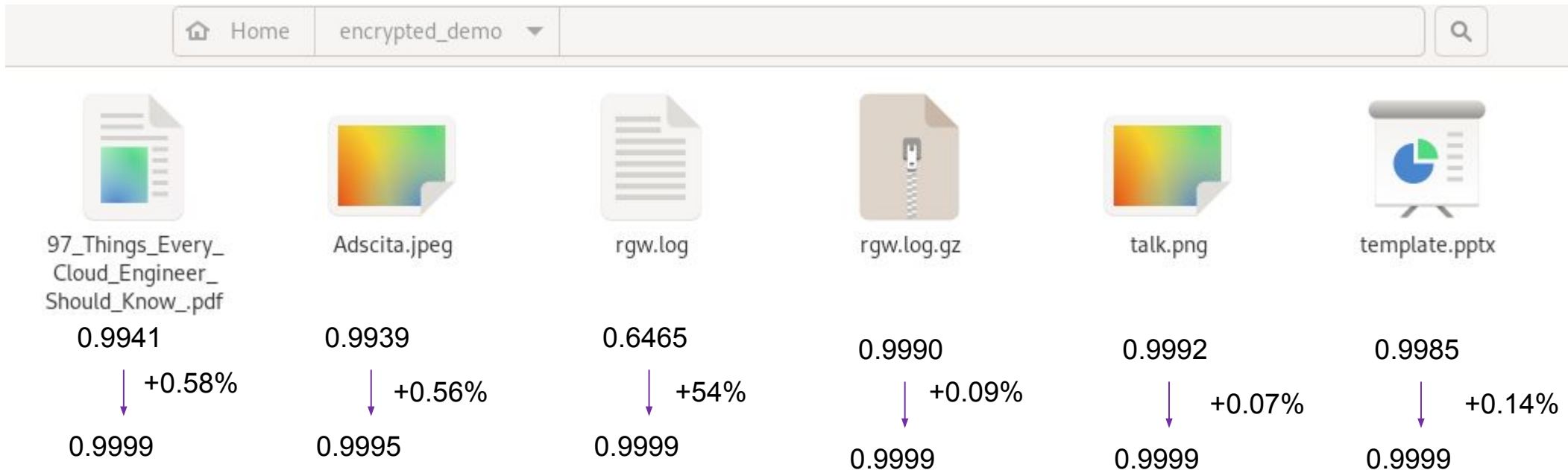
Entropy of a File

- Now the random variable is a file that contains bytes sampled from a 256 word alphabets
- The closer the distribution to a uniform one, the higher the entropy is
- Compressed files (JPEG, MPEG, MP3, ZIP, PDF...) will have higher entropy
- Encrypted files will have higher entropy
- File types are difficult to detect, and file name suffix are an unreliable mean of detection
- There is no good absolute entropy threshold for detecting when a file is encrypted

Home		demo					
							
97_Things_Every_Cloud_Engineer_Should_Know_.pdf		Adscita.jpeg		rgw.log		rgw.log.gz	
0.9941		0.9939		0.6465		0.9990	
							
				talk.png		template.pptx	
				0.9992		0.9985	

Encrypted Files

- **Good Encryption == Higher Entropy**, and ransomware wants good encryption
- **Per file entropy** have changed
- **Per directory**, the file entropy increased at certain rate



Ceph Storage System

Ceph is **Free and Open Source** Storage system

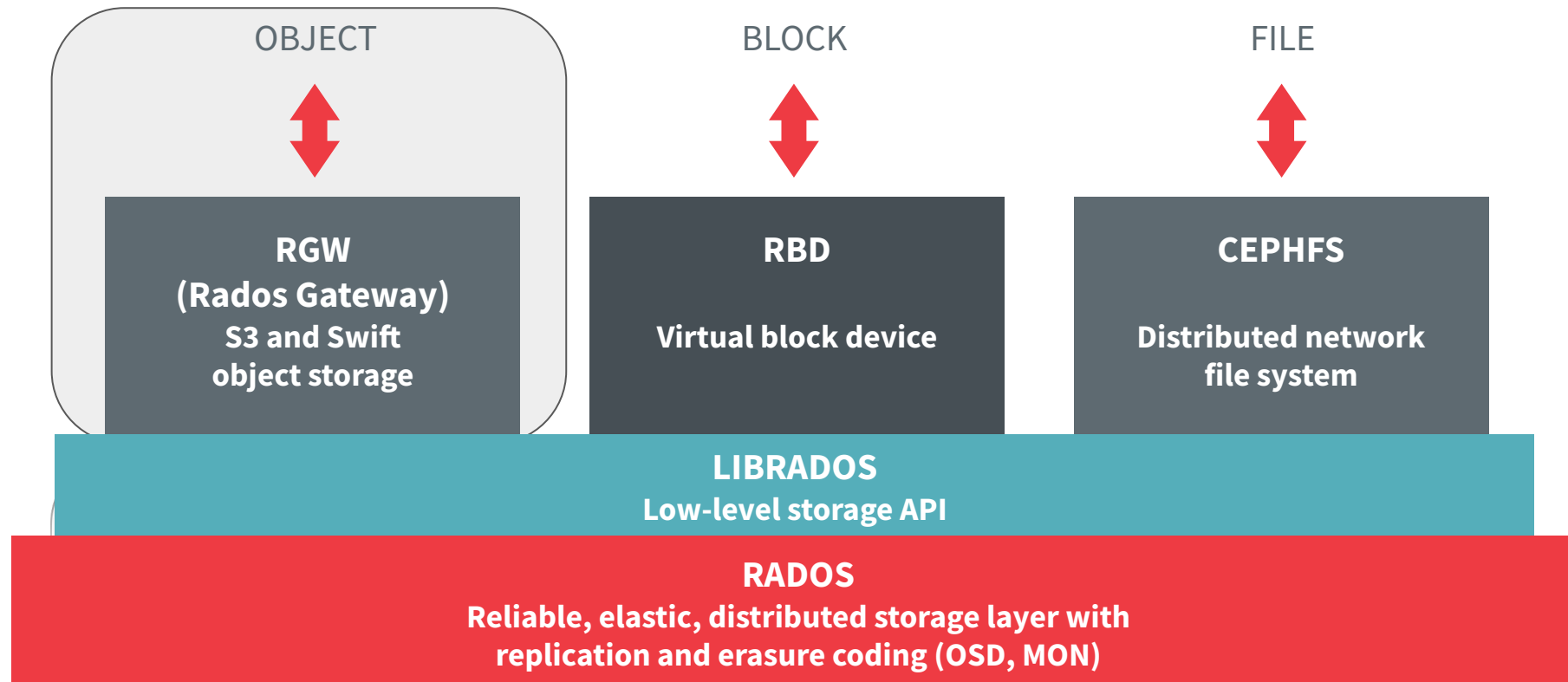
- Free to use (...as in beer)
- Free from vendor lock-in
 - Commodity servers
 - IP Networks
 - HDDs, SSDs, NVMe,...
- Open Source and Free to change (...as in speech)
- Ceph is also “Open Ended” with many integration points

B.Y.O.D



Ceph's Object Store - Rados Gateway (RGW)

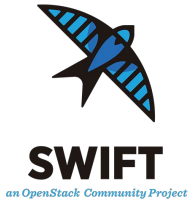
Ceph is a **Unified** Storage System for Object, Block and File



“Open Ended” Ceph/RGW



kafka



C++ Object Classes

Lua Object Classes

Bucket Notifications

Cloud Storage

RGW Lua Scripting!

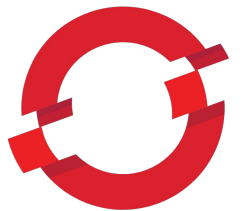


Rook - Cloud Native Storage Orchestration

- Operator for Kubernetes and OpenShift
- Makes deployment and configuration “easy as a YAML” for (almost) everything
- For Lua the “toolbox” pod should be used to run `radosgw-admin` commands

```
apiVersion: ceph.rook.io/v1
kind: CephObjectStore
metadata:
  name: my-store
  namespace: rook-ceph
spec:
  metadataPool:
    replicated:
      size: 3
  dataPool:
    replicated:
      size: 3
  preservePoolsOnDelete: false
  gateway:
    port: 80
    securePort: 443
    instances: 1
```

```
apiVersion: ceph.rook.io/v1
kind: CephCluster
metadata:
  name: my-cluster
  namespace: rook-ceph # namespace:cluster
spec:
  dataDirHostPath: /var/lib/rook
  cephVersion:
    image: quay.io/ceph/ceph:v17
    allowUnsupported: true
  mon:
    count: 1
    allowMultiplePerNode: true
  mgr:
    count: 1
    allowMultiplePerNode: true
  dashboard:
    enabled: true
    crashCollector:
      disable: true
  storage:
    useAllNodes: true
    useAllDevices: true
  healthCheck:
    daemonHealth:
      mon:
        interval: 45s
        timeout: 600s
    disruptionManagement:
      managePodBudgets: true
```



OPENSIFT



Lua Scripting on the RGW

Why?

- **Mature and Powerful**
- **Easy to Learn**
- **Lightweight**
- **Efficient Integration (zero copy)**
- **Flexible**
- **No need to know C++ and rebuild Ceph :-)**



read/write global RGW table
that persist across request
contexts

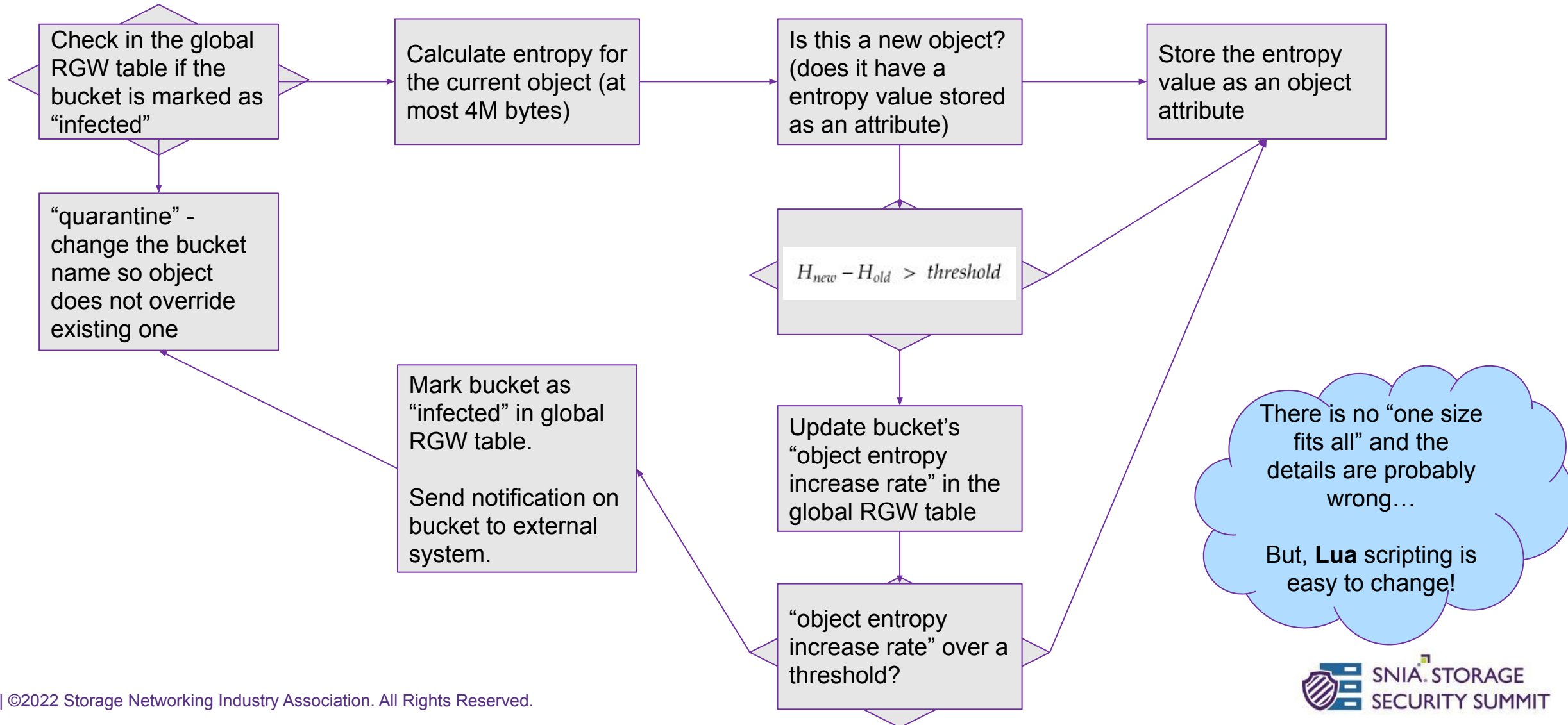
```
-- add metadata to objects that was not originally sent by the client
if Request.RGWOp == "put_obj" then
    Request.HTTP.Metadata[ "x-amz-meta-mydata" ] = RGW["mydata"]
    increment (RGW[Request.Bucket.Name.. "-metadata-added-count" ])
end
```

a table called
"Request".
points to the values of a
specific C++ struct
holding a request to the
RGW

Scripts and luarocks dependencies are managed using the `radosgw-admin` tool

(In Rook this will be done via the "toolbox pod")

Harnessing Object Storage and Lua



Resources

- SNIA Talk by Chris Lionetti on using storage behavioral patterns to detect ransomware:
 - <https://www.youtube.com/watch?v=WyEqyD2L4Xo>
- Papers on using entropy to detect ransomware:
 - <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8772046>
 - <https://arxiv.org/pdf/2106.14418.pdf>
- Lua scripting in Ceph Object Store:
 - <https://docs.ceph.com/en/quincy/radosgw/lua-scripting/>
- Ceph “tech talks” on Lua scripting:
 - <https://www.youtube.com/watch?v=anQJugs27hE>
 - <https://www.youtube.com/watch?v=F8zKFI60q9g>
- Object Store in Rook:
 - <https://rook.io/docs/rook/v1.9/ceph-object.html>
- Blog on setting Object Store in Ceph using Rook:
 - <https://shonpaz.medium.com/run-your-s3-object-storage-service-on-openshift-using-rook-ceph-35ca321af212>
- Demo Scripts:
 - entropy calculation in Lua: <https://gist.github.com/yuvalif/c1386ce3c722ad55f7b8a96ef2190662>
 - my “wannacry” script: <https://gist.github.com/yuvalif/74c8bb34406a8d6215446d46aaefe543>



Please take a moment to rate this session.

Your feedback is important to us.