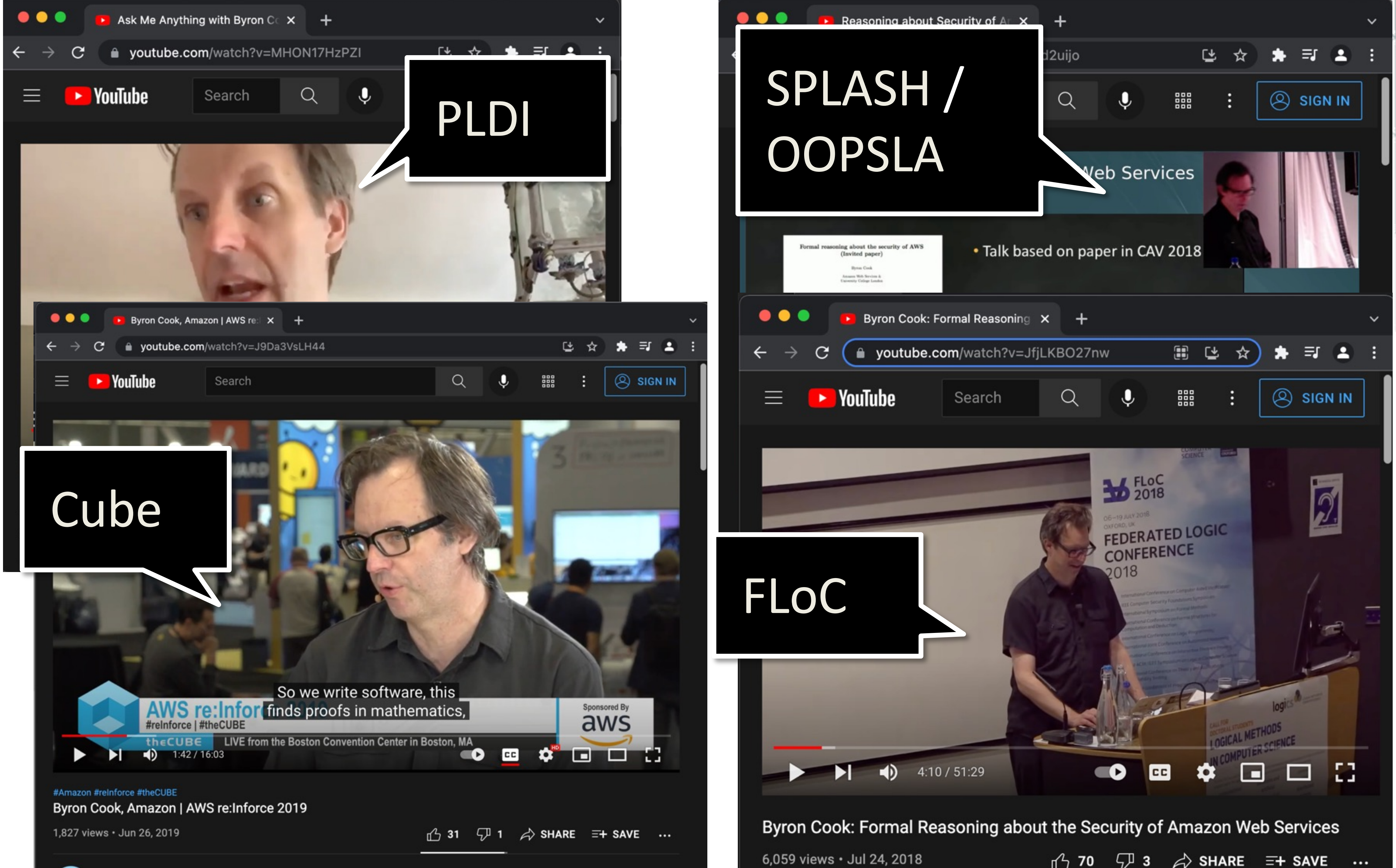
Automated Reasoning in and about the Cloud, plus Applications in Cryptography

Rod Chapman, Senior Principal Applied Scientist, AWS

amazon | science



Ask Me Anything with Byron C
youtube.com/watch?v=MHON17HzPZI
PLDI
Reasoning about Security of A
SPLASH / OOPSLA
Web Services
Formal reasoning about the security of AWS (Invited paper)
Talk based on paper in CAV 2018
Byron Cook, Amazon | AWS re:
youtube.com/watch?v=J9Da3VsLH44
Cube
So we write software, this finds proofs in mathematics,
AWS re:Inforce
#reInforce | #theCUBE
LIVE from the Boston Convention Center in Boston, MA
Sponsored By aws
1:42 / 16:03
#Amazon #reInforce #theCUBE
Byron Cook, Amazon | AWS re:Inforce 2019
1,827 views • Jun 26, 2019
Byron Cook: Formal Reasoning
youtube.com/watch?v=JfjLKBO27nw
FLoC 2018
FEDERATED LOGIC CONFERENCE 2018
FLoC
4:10 / 51:29
Byron Cook: Formal Reasoning about the Security of Amazon Web Services
6,059 views • Jul 24, 2018

AR Success at Amazon…

How come AR has been successful at Amazon?

Some possible answers…

1. Trust

   The ability to make justified and *universal* claims of correctness for our users' applications, their data, and our infrastructure…

   Universal? "*For all* users, *for all* buckets, *for all* requests, *for all* possible VPCs, property P is true…"

AR Success at Amazon…

How come AR has been successful at Amazon?

Some possible answers…

1. Trust (continued…)

   AR also produces *finite* proofs about *infinite* systems.

   Example: Pythagoras' theorem. A short proof (well…several proofs) about an infinite number of right-angle triangles…

   The catch: *soundness* of verification requires discipline, determination and persistence…

AR Success at Amazon…

How come AR has been successful at Amazon?

Some possible answers…

2. Scale

At "Cloud Scale" there are no corner-cases…
Verification by "Appeal to unlikely things not happening" does not work!

Example: In 2021, Amazon Simple Storage Service (S3) was responding to more than 10 Million requests…

…per second.

More recent Scale metrics

Supporting “Prime Day” 2023…

- Amazon Elastic Block Store (EBS) handled 15.35 Trillion requests and 764 Petabytes of data transfer *per day*…
- Amazon DynamoDB peaked at 126 Million request *per second*…
- Amazon CloudFront handled a peak load of over 500 Million HTTP requests *per minute*.
- Amazon Simple Queue Service (SQS) peaked at 86 Million requests *per second*.

(Data from https://aws.amazon.com/blogs/aws/prime-day-2023-powered-by-aws-all-the-numbers/)

AR Success at Amazon…

How come AR has been successful at Amazon?

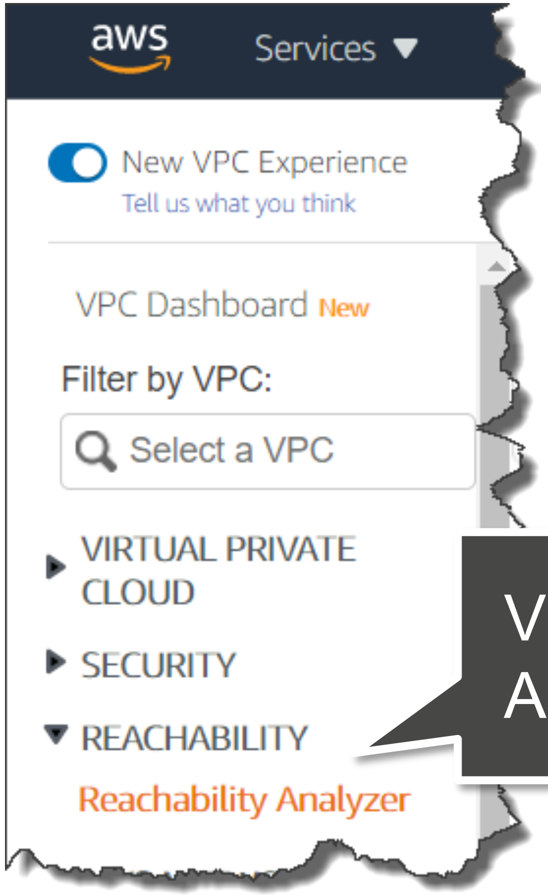
Some possible answers…

3. Listening to Customers…

   Don't try to verify "Everything…"

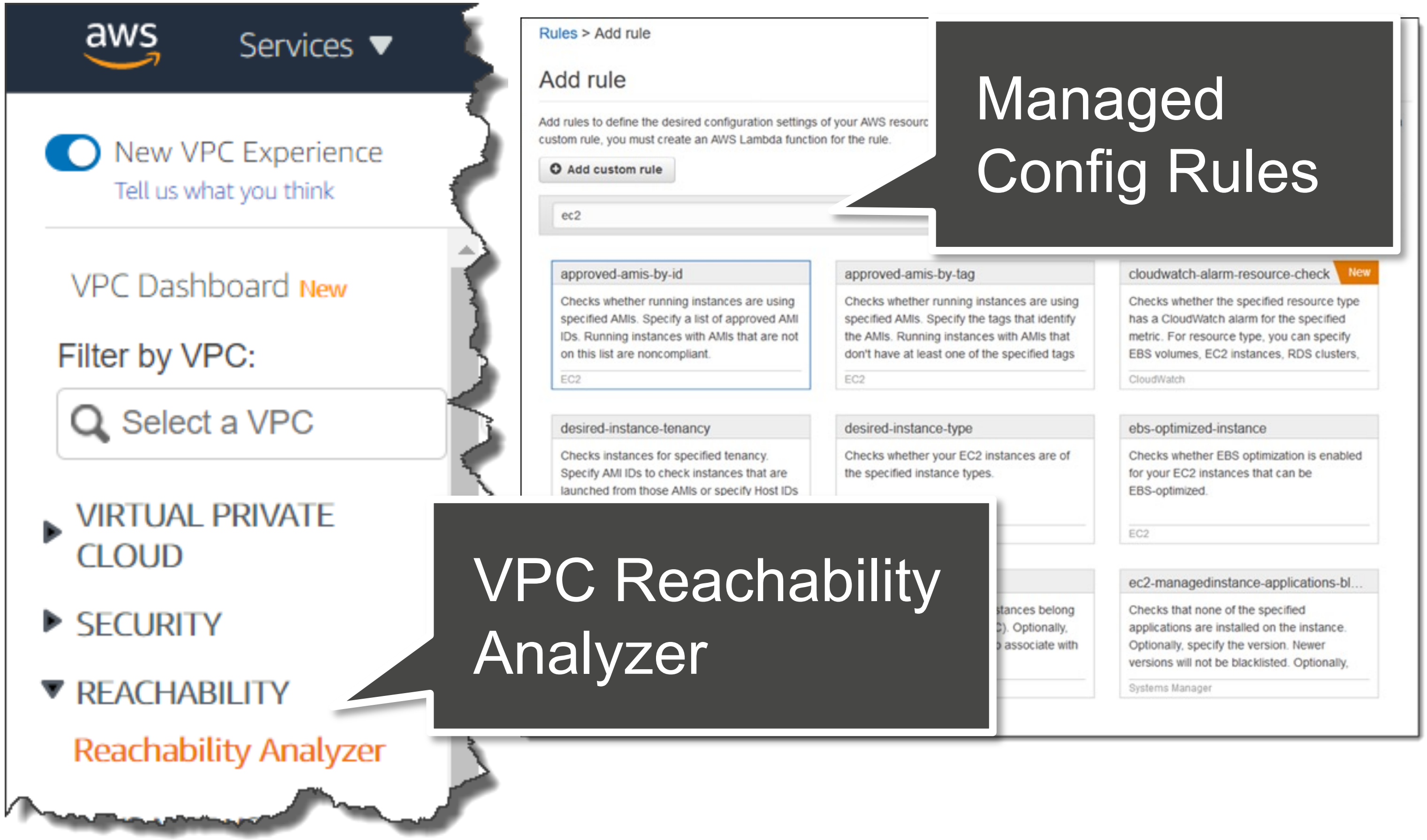
   Listen to customers, and concentrate on what they care most about.

   Verify *properties* (but don't sacrifice soundness…)

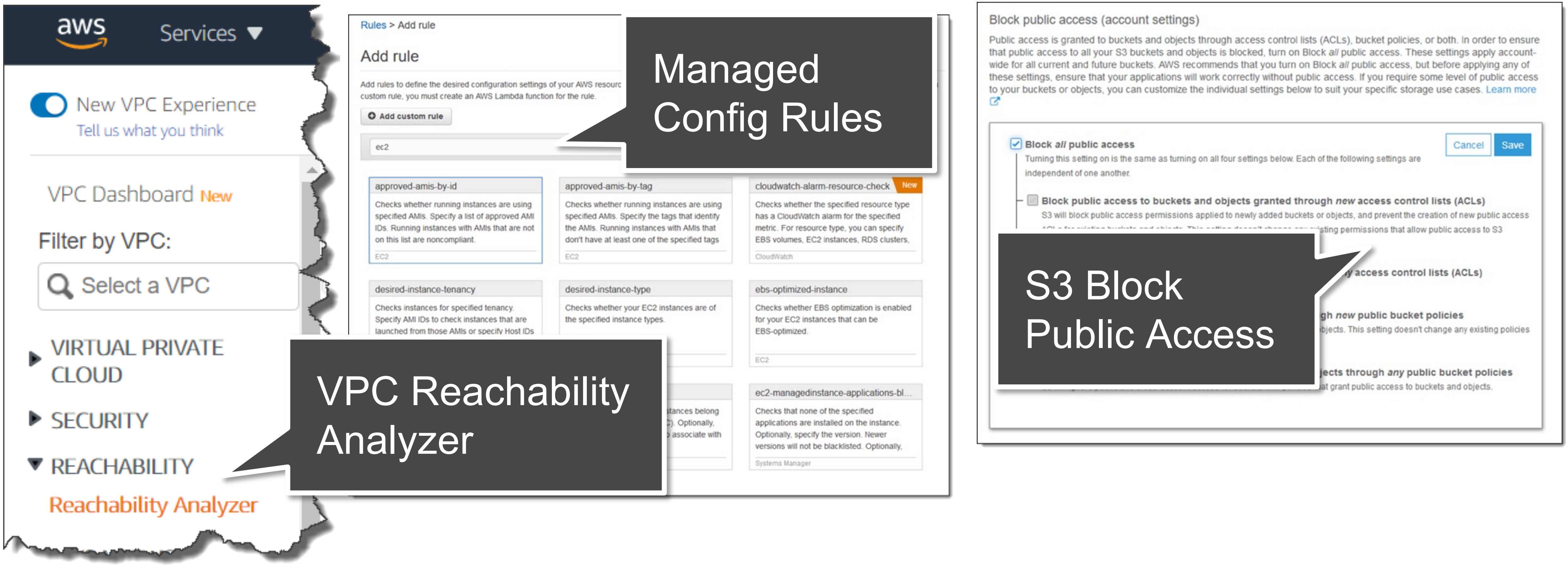


VPC Reachability Analyzer

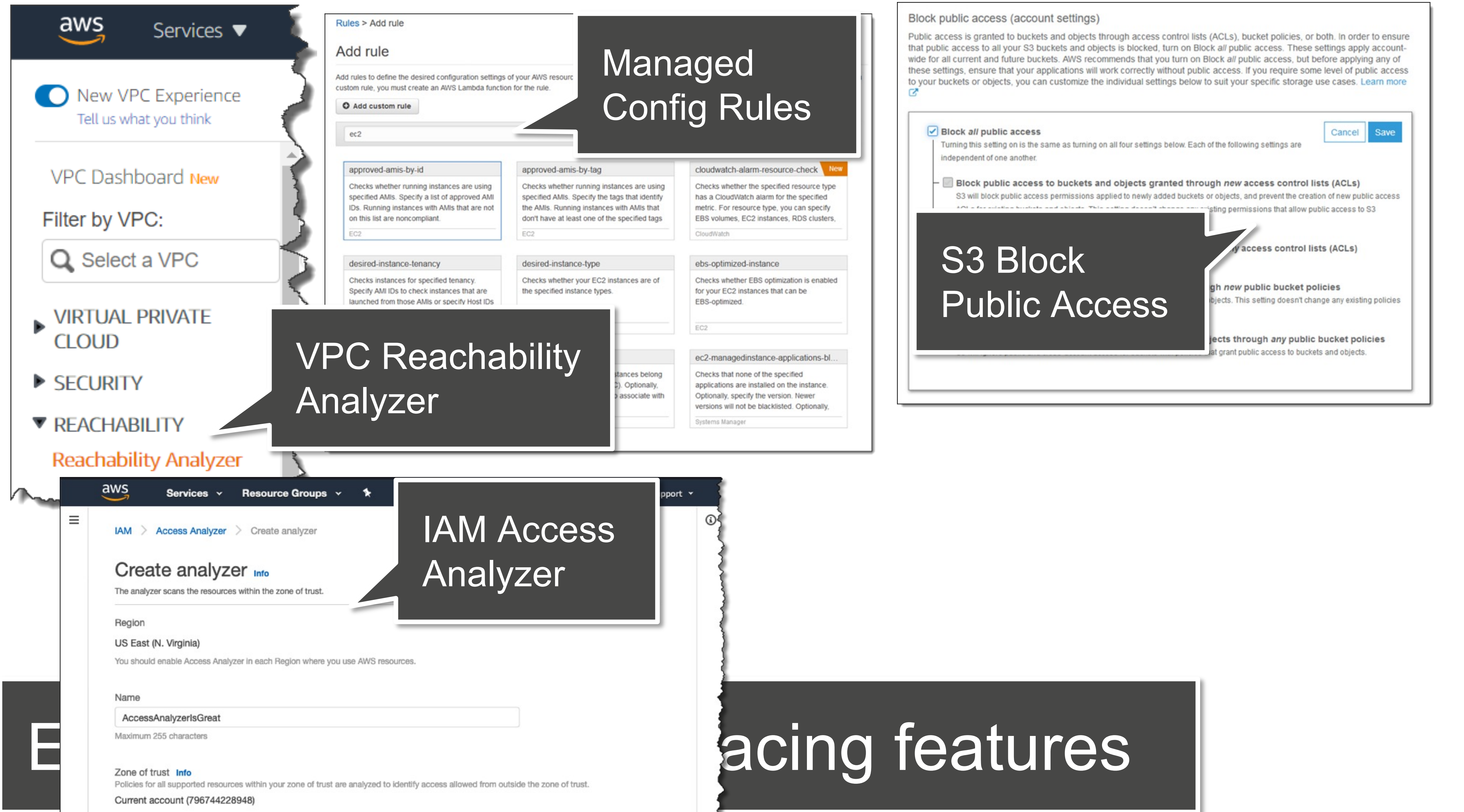
Examples of customer-facing features

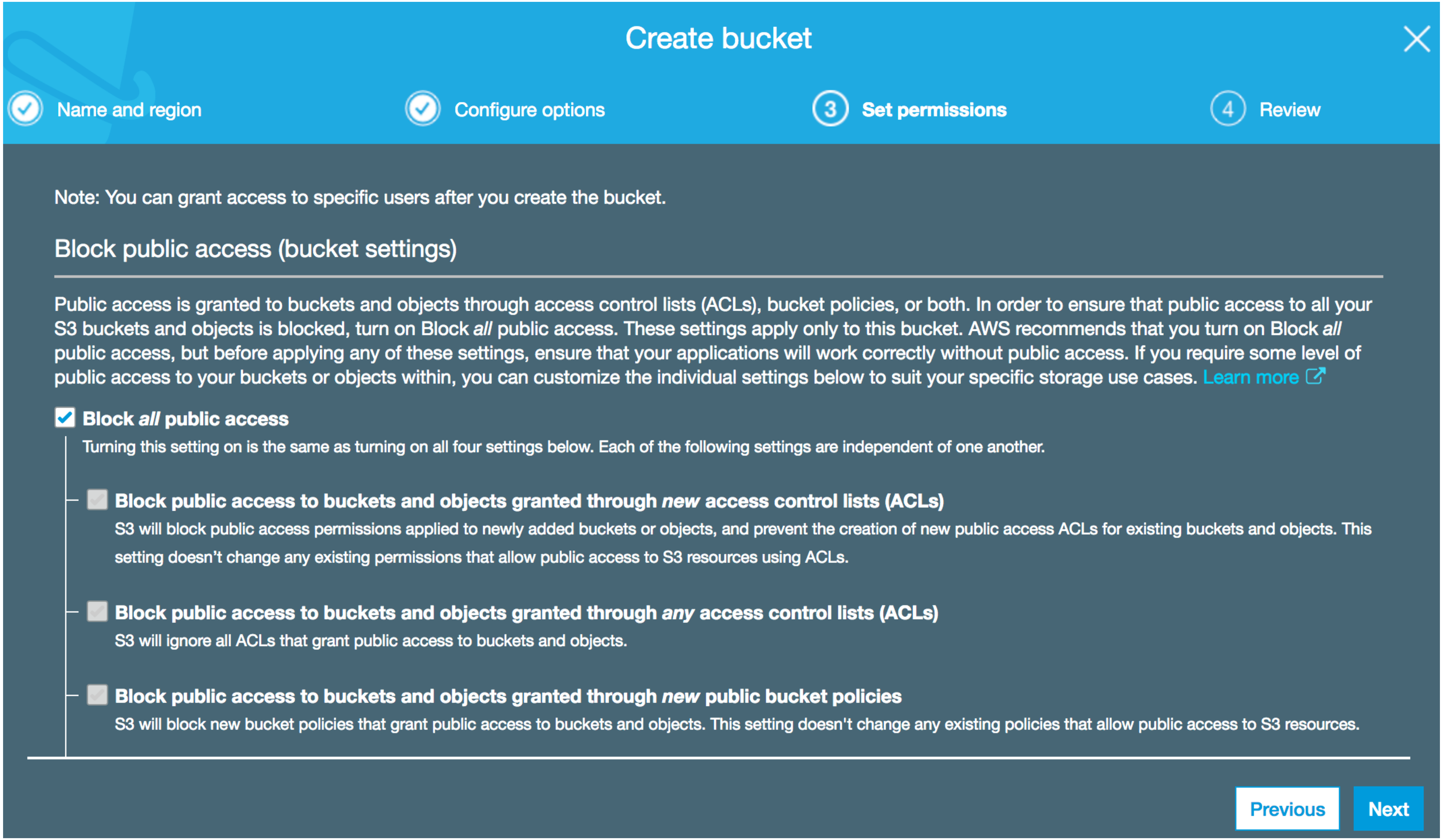


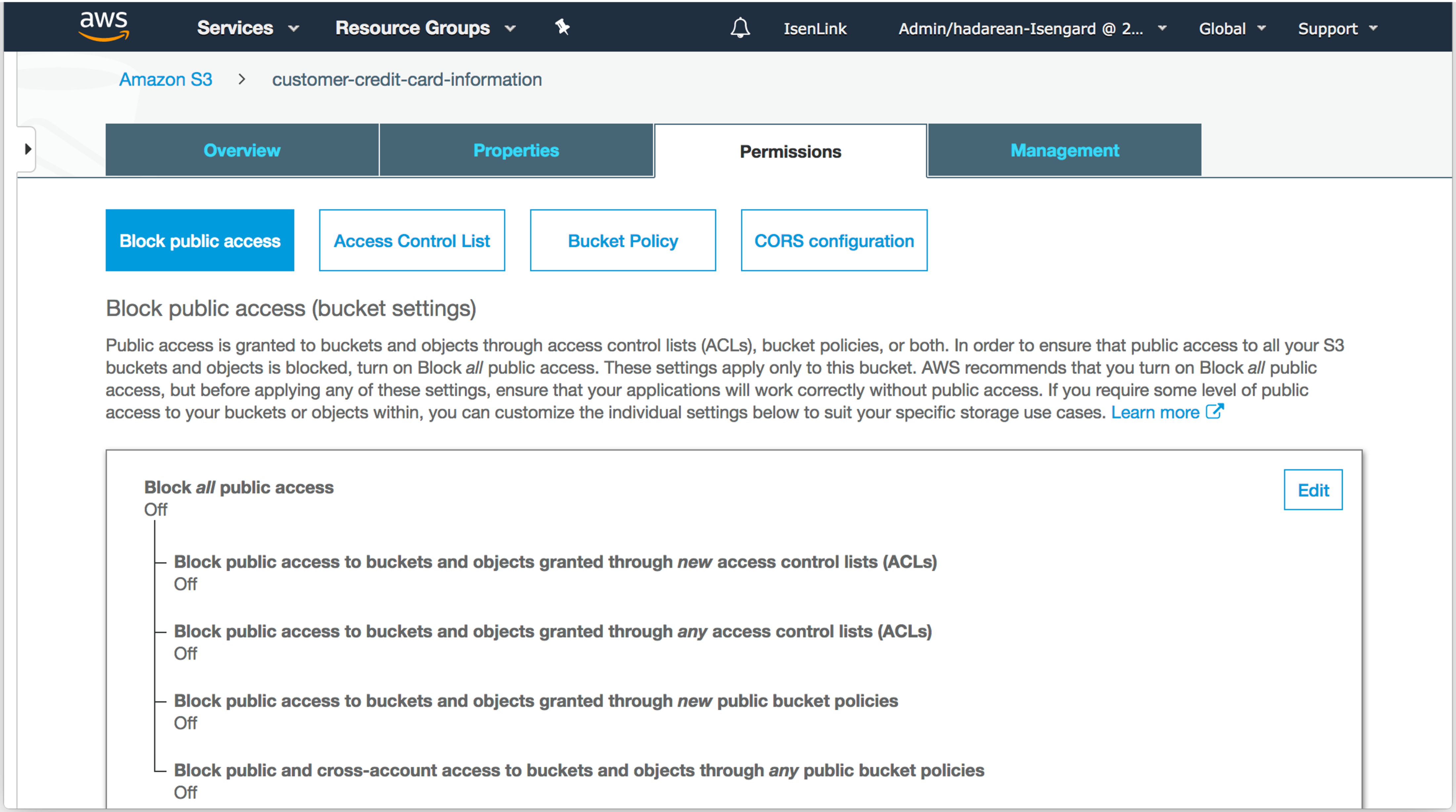
Examples of customer-facing features

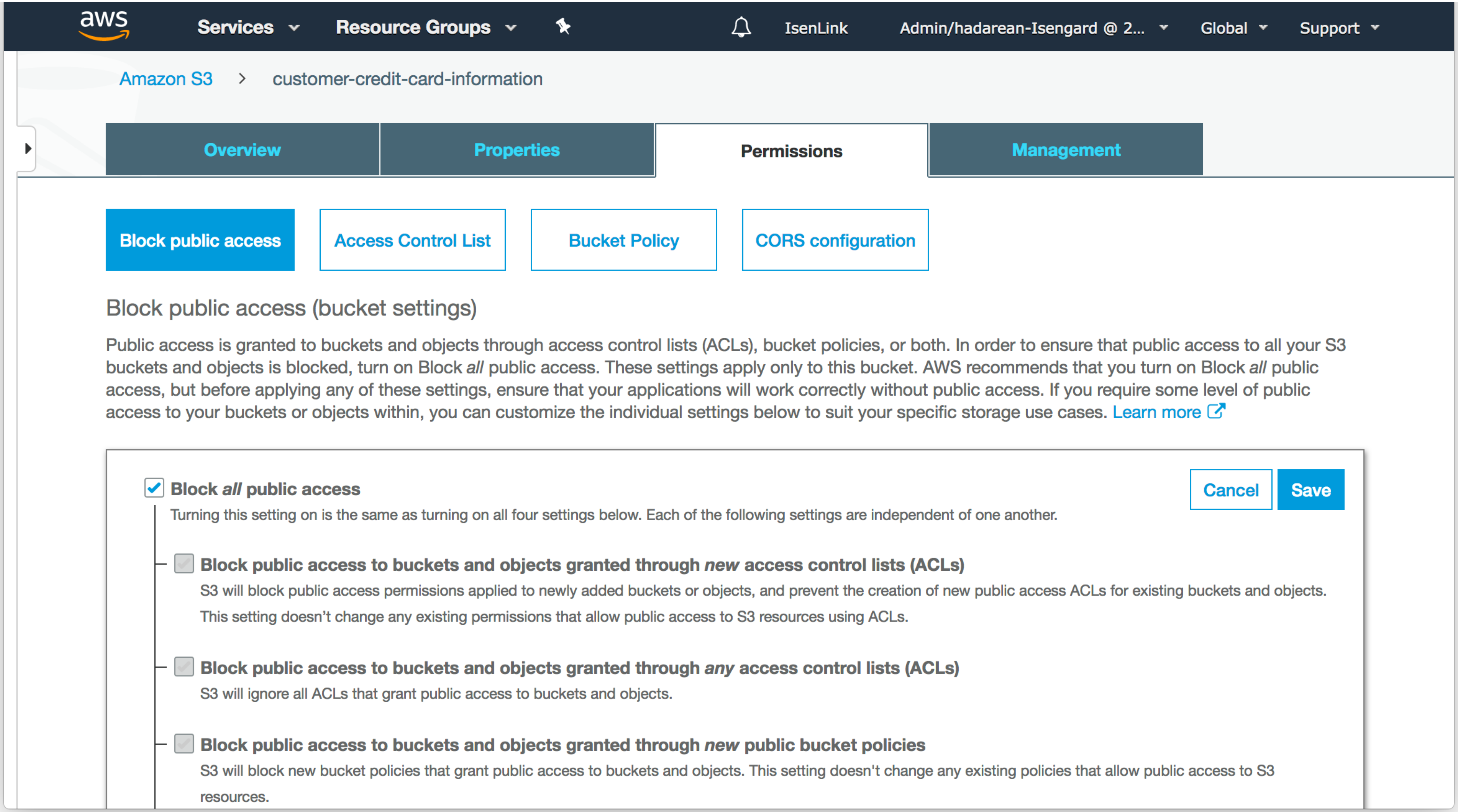


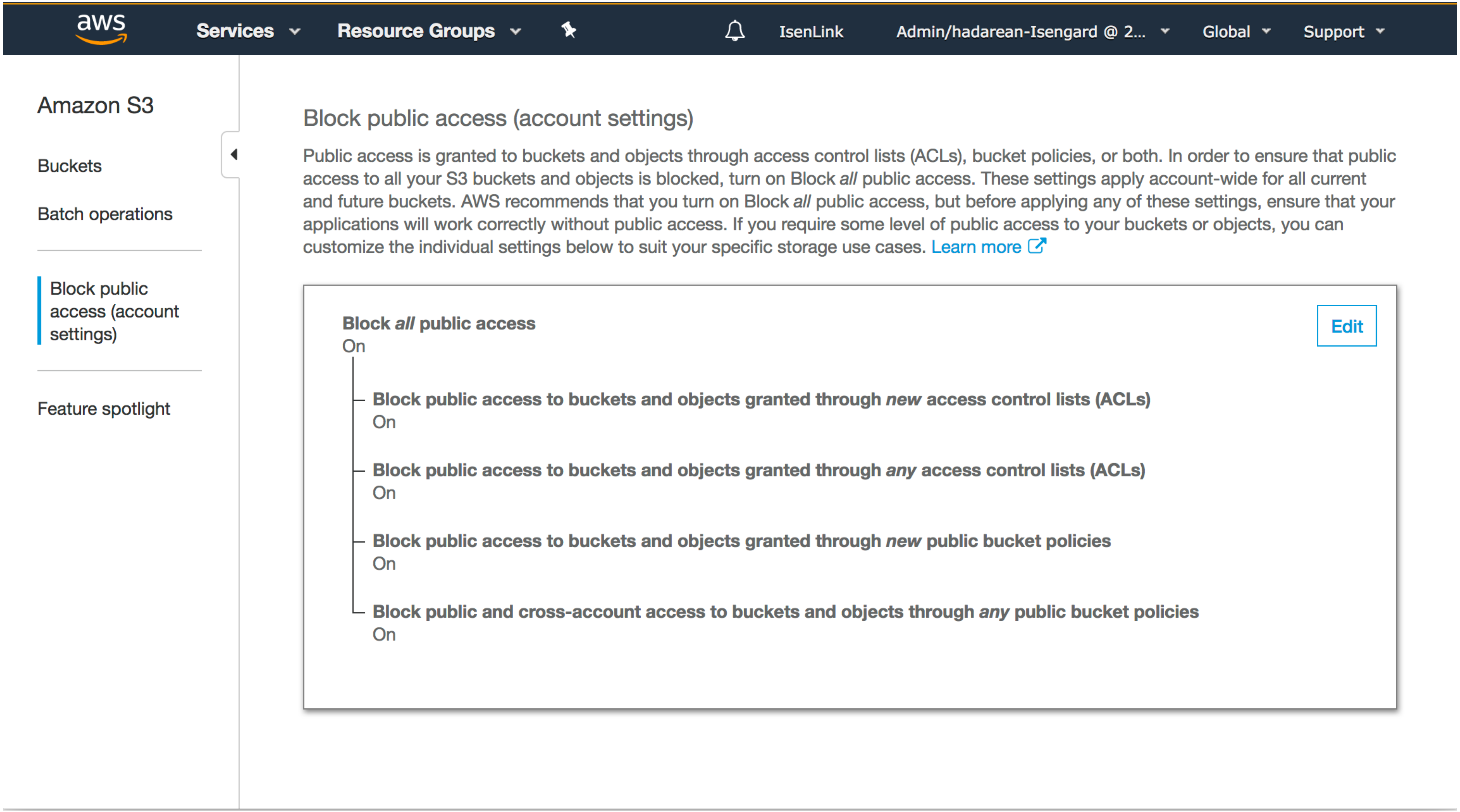
Examples of customer-facing features


aws
Services
New VPC Experience
Tell us what you think
VPC Dashboard New
Filter by VPC:
Select a VPC
VIRTUAL PRIVATE CLOUD
SECURITY
REACHABILITY
Reachability Analyzer
VPC Reachability Analyzer
Rules > Add rule
Add rule
Add custom rule
ec2
approved-amis-by-id
approved-amis-by-tag
cloudwatch-alarm-resource-check
desired-instance-tenancy
desired-instance-type
ebs-optimized-instance
ec2-managedinstance-applications-bl...
Managed Config Rules
Block public access (account settings)
Block all public access
Cancel
Save
S3 Block Public Access
Services
Resource Groups
IAM > Access Analyzer > Create analyzer
Create analyzer Info
The analyzer scans the resources within the zone of trust.
Region
US East (N. Virginia)
You should enable Access Analyzer in each Region where you use AWS resources.
Name
AccessAnalyzerIsGreat
Maximum 255 characters
Zone of trust Info
Policies for all supported resources within your zone of trust are analyzed to identify access allowed from outside the zone of trust.
Current account (796744228948)
IAM Access Analyzer
E
acing features


Create bucket
Name and region
Configure options
3 Set permissions
4 Review
Note: You can grant access to specific users after you create the bucket.
Block public access (bucket settings)
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, or both. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block *all* public access. These settings apply only to this bucket. AWS recommends that you turn on Block *all* public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more
Block *all* public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
Block public access to buckets and objects granted through *new* access control lists (ACLs)
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn’t change any existing permissions that allow public access to S3 resources using ACLs.
Block public access to buckets and objects granted through *any* access control lists (ACLs)
S3 will ignore all ACLs that grant public access to buckets and objects.
Block public access to buckets and objects granted through *new* public bucket policies
S3 will block new bucket policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
Previous
Next


aws
Services
Resource Groups
IsenLink
Admin/hadarean-Isengard @ 2...
Global
Support
Amazon S3
customer-credit-card-information
Overview
Properties
Permissions
Management
Block public access
Access Control List
Bucket Policy
CORS configuration
Block public access (bucket settings)
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, or both. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more
Block all public access
Off
Edit
Block public access to buckets and objects granted through new access control lists (ACLs)
Off
Block public access to buckets and objects granted through any access control lists (ACLs)
Off
Block public access to buckets and objects granted through new public bucket policies
Off
Block public and cross-account access to buckets and objects through any public bucket policies
Off

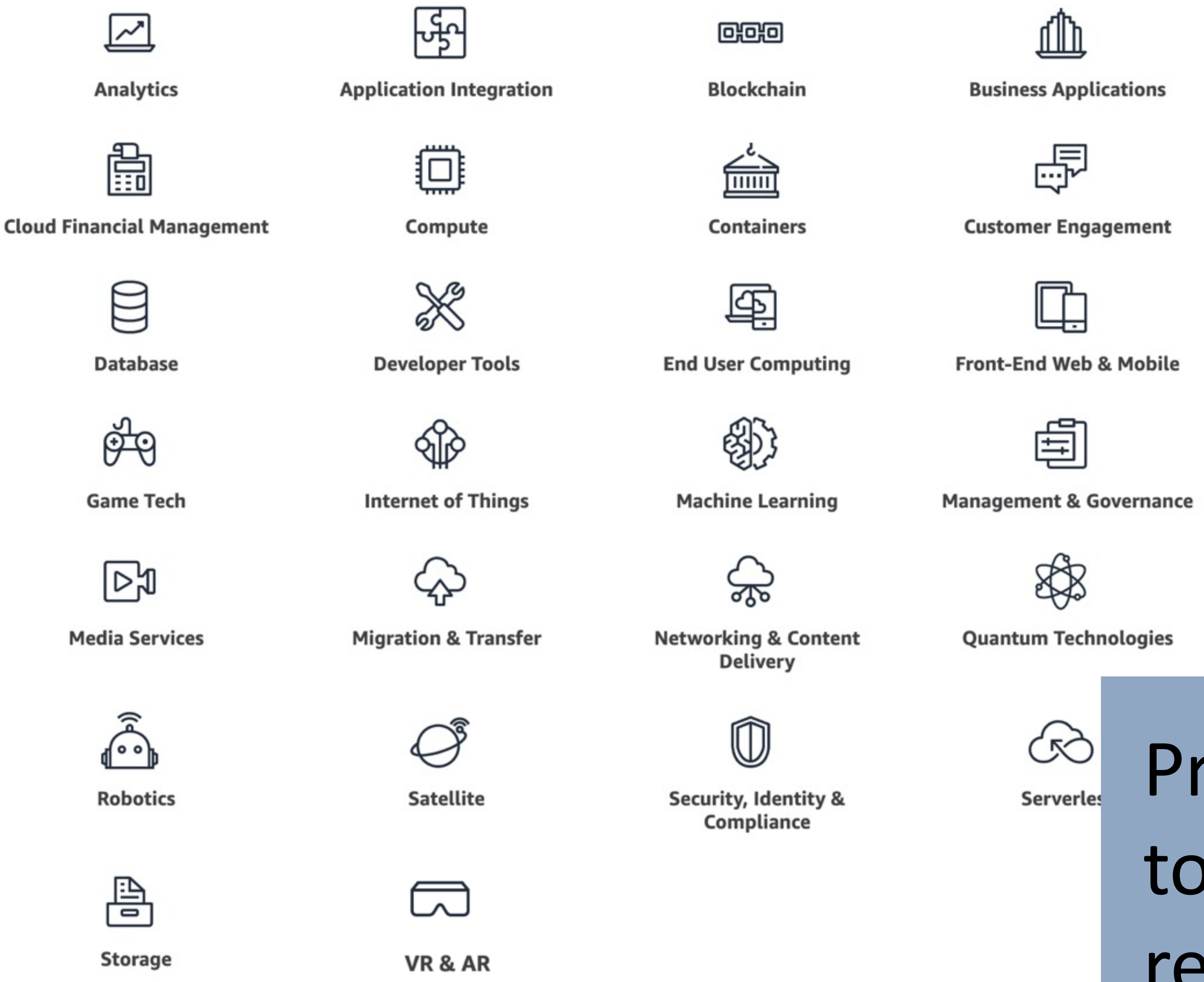

aws
Services
Resource Groups
IsenLink
Admin/hadarean-Isengard @ 2...
Global
Support
Amazon S3
customer-credit-card-information
Overview
Properties
Permissions
Management
Block public access
Access Control List
Bucket Policy
CORS configuration
Block public access (bucket settings)
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, or both. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more
Block all public access
Cancel
Save
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
Block public access to buckets and objects granted through new access control lists (ACLs)
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
Block public access to buckets and objects granted through any access control lists (ACLs)
S3 will ignore all ACLs that grant public access to buckets and objects.
Block public access to buckets and objects granted through new public bucket policies
S3 will block new bucket policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.


aws
Services
Resource Groups
IsenLink
Admin/hadarean-Isengard @ 2...
Global
Support
Amazon S3
Buckets
Batch operations
Block public access (account settings)
Feature spotlight
Block public access (account settings)
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, or both. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply account-wide for all current and future buckets. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects, you can customize the individual settings below to suit your specific storage use cases. Learn more
Block all public access
On
Edit
Block public access to buckets and objects granted through new access control lists (ACLs)
On
Block public access to buckets and objects granted through any access control lists (ACLs)
On
Block public access to buckets and objects granted through new public bucket policies
On
Block public and cross-account access to buckets and objects through any public bucket policies
On

AWS product categories

- Analytics
- Application Integration
- Blockchain
- Business Applications
- Cloud Financial Management
- Compute
- Containers
- Customer Engagement
- Database
- Developer Tools
- End User Computing
- Front-End Web & Mobile
- Game Tech
- Internet of Things
- Machine Learning
- Management & Governance
- Media Services
- Migration & Transfer
- Networking & Content Delivery
- Quantum Technologies
- Robotics
- Satellite
- Security, Identity & Compliance
- Serverless
- Storage
- VR & AR

AWS product categories



Practically every area touched by automated reasoning in some way


Scientists at Principal level or above
View Badge Photo
Rajeev Joshi
Gustavo Petri
Dominic Mulligan
Emina Torlak
Tancrède Lepoint
View Badge Photo
Jim Grundy
Muhammad Naveed
Cezara Drăgoi
Leo de Moura
Aws Albarghouthi
View Badge Photo
Jared Davis
John Harrison
Willem Visser
Zvonimir Rakamaric
Cesar Munoz
Byron Cook
View Badge Photo
Jaco Geldenhuys
Mike Whalen
Andrew Gacek
Murat Demirbas
Ben Liblit
Mark Tuttle
Nathan Chong
View Custom Photo
Ernie Cohen
Mike Hicks
Sean McLaughlin
Remi Delmas
Rustan Leino
Serdar Tasiran
Bruno Dutertre
Temesghen Kahsai
Daniel Kroening
Dimitra Giannakopoulou
Lee Pike
John Backes
Rod Chapman

Dafny

kani

CVC5

AWS Zelkova

THE LINUX FOUNDATION
TLA+
FOUNDATION

AMZN Dust

s2n

AWS Shuttle

P

AWS Tiros

CBMC

Z3

CEDAR

AR and Cryptography

AR and Cryptography

Let's concentrate on Software and its verification for now…

How come crypto software is so hard to get right?

1. Correctness?
2. Performance? (owing to scale, this means $$$)
3. Side-Channel freedom?
4. Longevity?

"All of the above please!"

AR and Cryptography

Emerging Trends…

Things I've noticed in the last few years…

1. "Formal is Normal"

   Crypto research papers proposing new thing X come with mathematical proofs of X's security properties…

   At many levels… pure Math stuff, security properties, formal specification languages, protocols, code…

   Will future RFCs have formal specification as the one and only notation?

AR and Cryptography

Emerging Trends…

Things I've noticed in the last few years…

2. "Joined up Formal"

Refinement proofs? Verified compilation?

"Joins up" proofs of math stuff, with protocols, and code…

Problem: where to "draw the line" around the formal model?
HL programming language, ISA, micro-arch, transistors…

AR and Cryptography

Emerging Trends…

Things I've noticed in the last few years…

3. "Formal and Fast" not "Formal xor Fast"

Traditional myth: "Formal is Slow…"

Or… put another way… "Our production code has to go really really fast (and run on bare-metal), so we can only use C, C, C, any language that starts with C, or assembly if we're really desperate…"

"Formal and Fast"???

Introducing… s2n-bignum.

README.md

s2n-bignum

This is a collection of bignum arithmetic routines designed for cryptographic applications. All routines are written in pure machine code, designed to be callable from C and other high-level languages, with separate but API-compatible versions of each function for 64-bit x86 (x86_64) and ARM (aarch64). Each function is written in a constant-time style to avoid timing side-channels, and is accompanied by a machine-checked formal proof that its mathematical result is correct, based on a formal model of the underlying machine.

s2n-bignum – what do I get?

- Functions for elliptic curve field elements, point operations, and (point x scalar) multiplication…
- …for each of the curves NIST P-256, P-384, P-521, plus secp256k1, SM2, and curve25519…
- AND… primitive operations (e.g. a^b mod n) for RSA cryptosystem with 2048, 3072, and 4096 bit modulus n.

s2n-bignum – what do I get?

- …implemented for x86_64 and ARM64, for at least 2 micro-architectures each…
- … with proofs of functional correctness in HOL-Lite…
- … with Apache-2.0 or ISC licence
- Available at https://github.com/awslabs/s2n-bignum

s2n-bignum – for example…

- in include/s2n-bignum.h

```
// Add modulo p_25519, z := (x + y) mod p_25519,
// assuming x and y reduced
// Inputs x[4], y[4]; output z[4]

extern void bignum_add_p25519 (uint64_t z[static 4],
                               uint64_t x[static 4],
                               uint64_t y[static 4]);
```

s2n-bignum – for example…

- in arm/curve25519/bignum_add_p25519.S

```
// Add as [d3; d2; d1; d0] = x + y; since we assume
// x, y < 2^255 - 19 this sum fits in 256 bits

        ldp     d0, d1, [x]
        ldp     c0, c1, [y]
        adds    d0, d0, c0
        adcs    d1, d1, c1
        ldp     d2, d3, [x, #16]
        ldp     c0, c1, [y, #16]
        adcs    d2, d2, c0
        adc     d3, d3, c1

// Now x+y >= 2^255 - 19 <=> x+y+(2^255+19) >= 2^256
// Form [c3; c2; c1; c0] = (x+y) + (2^255+19), with CF
// for the comparison

        mov     c3, #0x8000000000000000
        adds    c0, d0, #19
        adcs    c1, d1, xzr
        adcs    c2, d2, xzr
        adcs    c3, d3, c3
```

```
// If the comparison holds, select [c3; c2; c1; c0].
// There's no need to mask it since in this case it
// is ((x + y) + (2^255 + 19)) - 2^256 because the
// top carry is lost, which is the
// desired (x + y) - (2^255 - 19).

        csel    d0, d0, c0, cc
        csel    d1, d1, c1, cc
        csel    d2, d2, c2, cc
        csel    d3, d3, c3, cc

// Store the result

        stp     d0, d1, [z]
        stp     d2, d3, [z, #16]

        ret
```

s2n-bignum – for example…

- in arm/proofs/bignum_add_p25519.ml

```
let p_25519 = new_definition `p_25519 = 57896044618658097711785492504343953926634992332820282019728792003956564819949`;;

let BIGNUM_ADD_P25519_CORRECT = time prove
 (`!z x y m n pc.
        nonoverlapping (word pc,0x50) (z,8 * 4)
        ==> ensures arm
             (\s. aligned_bytes_loaded s (word pc) bignum_add_p25519_mc /\
                  read PC s = word pc /\
                  C_ARGUMENTS [z; x; y] s /\
                  bignum_from_memory (x,4) s = m /\
                  bignum_from_memory (y,4) s = n)
             (\s. read PC s = word (pc + 0x4c) /\
                  (m < p_25519 /\ n < p_25519
                   ==> bignum_from_memory (z,4) s = (m + n) MOD p_25519))
           (MAYCHANGE [PC; X3; X4; X5; X6; X7; X8; X9; X10] ,,
            MAYCHANGE SOME_FLAGS ,,
            MAYCHANGE [memory :> bignum(z,4)])`,
```

(Plus many lines deleted…)

s2n-bignum – for example…

- in arm/proofs/bignum_add_p25519.ml

```
let p_25519 = new_definition `p_25519 =
57896044618658097711785492504343953926634992
332820282019728792003956564819949`;;
```

```
let BIGNUM_ADD_P25519_CORRECT = time prove
 (`!z x y m n pc.
        nonoverlapping (word pc,0x50) (z,8 * 4)
        ==> ensures arm
             (\s. aligned_bytes_loaded s (word pc) bignum_add_p25519_mc /\
                  read PC s = word pc /\
                  C_ARGUMENTS [z; x; y] s /\
                  bignum_from_memory (x,4) s = m /\
                  bignum_from_memory (y,4) s = n)
             (\s. read PC s = word (pc + 0x4c) /\
                  (m < p_25519 /\ n < p_25519
                   ==> bignum_from_memory (z,4) s = (m + n) MOD p_25519))
          (MAYCHANGE [PC; X3; X4; X5; X6; X7; X8; X9; X10] ,,
           MAYCHANGE SOME_FLAGS ,,
           MAYCHANGE [memory :> bignum(z,4)])`,
```

(Plus many lines deleted…)

s2n-bignum – for example…

- in arm/proofs/bignum_add_p

Read 4 64-bit words from memory address x, and interpret as a little-endian Integer m. Same for y and n

```
let p_25519 = new_definition `p_25519 = 578960446186580 … 3956564819949`;;

let BIGNUM_ADD_P25519_CORRECT = time prove
 (`!z x y m n pc.
      nonoverlapping (word pc,0x50) (z,8 * 4)
      ==> ensures arm
          (\s. aligned_bytes_loaded s (word pc) bignum_a
               read PC s = word pc /\
               C_ARGUMENTS [z; x; y] s /\
               bignum_from_memory (x,4) s = m /\
               bignum_from_memory (y,4) s = n)
          (\s. read PC s = word (pc + 0x4c) /\
               (m < p_25519 /\ n < p_25519
                ==> bignum_from_memory (z,4) s = (m + n) MOD p_25519))
        (MAYCHANGE [PC; X3; X4; X5; X6; X7; X8; X9; X10] ,,
         MAYCHANGE SOME_FLAGS ,,
         MAYCHANGE [memory :> bignum(z,4)])`,

(Plus many lines deleted…)
```

s2n-bignum – for example…

- in arm/proofs/bignum_add_p25519.ml

```
let p_25519 = new_definition `p_25519 = 57896044618658097711785492504343953926634992332820282019728792003956564819949`;;

let BIGNUM_ADD_P25519_CORRECT = time prove
 (`!z x y m n pc.
      nonoverlapping (word pc,0x50) (z,8 * 4)
      ==> ensures arm
          (\s. aligned_bytes_loaded s (word pc) bignum_add_p
               read PC s = word pc /\
               C_ARGUMENTS [z; x; y] s /\
               bignum_from_memory (x,4) s = m /\
               bignum_from_memory (y,4) s = n)
          (\s. read PC s = word (pc + 0x4c) /\
               (m < p_25519 /\ n < p_25519
                ==> bignum_from_memory (z,4) s = (m + n) MOD p_25519))
         (MAYCHANGE [PC; X3; X4; X5; X6; X7; X8; X9; X10] ,,
          MAYCHANGE SOME_FLAGS ,,
          MAYCHANGE [memory :> bignum(z,4)])`,
```

Precondition: if m and n and both reduced modulo p…

(Plus many lines deleted…)

s2n-bignum – for example…

- in arm/proofs/bignum_add_p25519.ml

```
let p_25519 = new_definition `p_25519 = 57896044618658097711785492504343953926634992332820282019728792003956564819949`;;

let BIGNUM_ADD_P25519_CORRECT = time prove
 (`!z x y m n pc.
      nonoverlapping (word pc,0x50) (z,8 * 4)
      ==> ensures arm
          (\s. aligned_bytes_loaded s (word pc) bignum_add_p25519_mc /\
               read PC s = word pc /\
               C_ARGUMENTS [z; x; y] s /\
               bignum_from_memory (x,4) s = m /\
               bignum_from_memory (y,4) s = n)
          (\s. read PC s = word (pc + 0x4c) /\
               (m < p_25519 /\ n < p_25519
                ==> bignum_from_memory (z,4) s = (m + n) MOD p_25519))
        (MAYCHANGE [PC; X3; X4; X5; X6; X7; X8; X9; X10] ,,
         MAYCHANGE SOME_FLAGS ,,
         MAYCHANGE [memory :> bignum(z,4)])`,
```

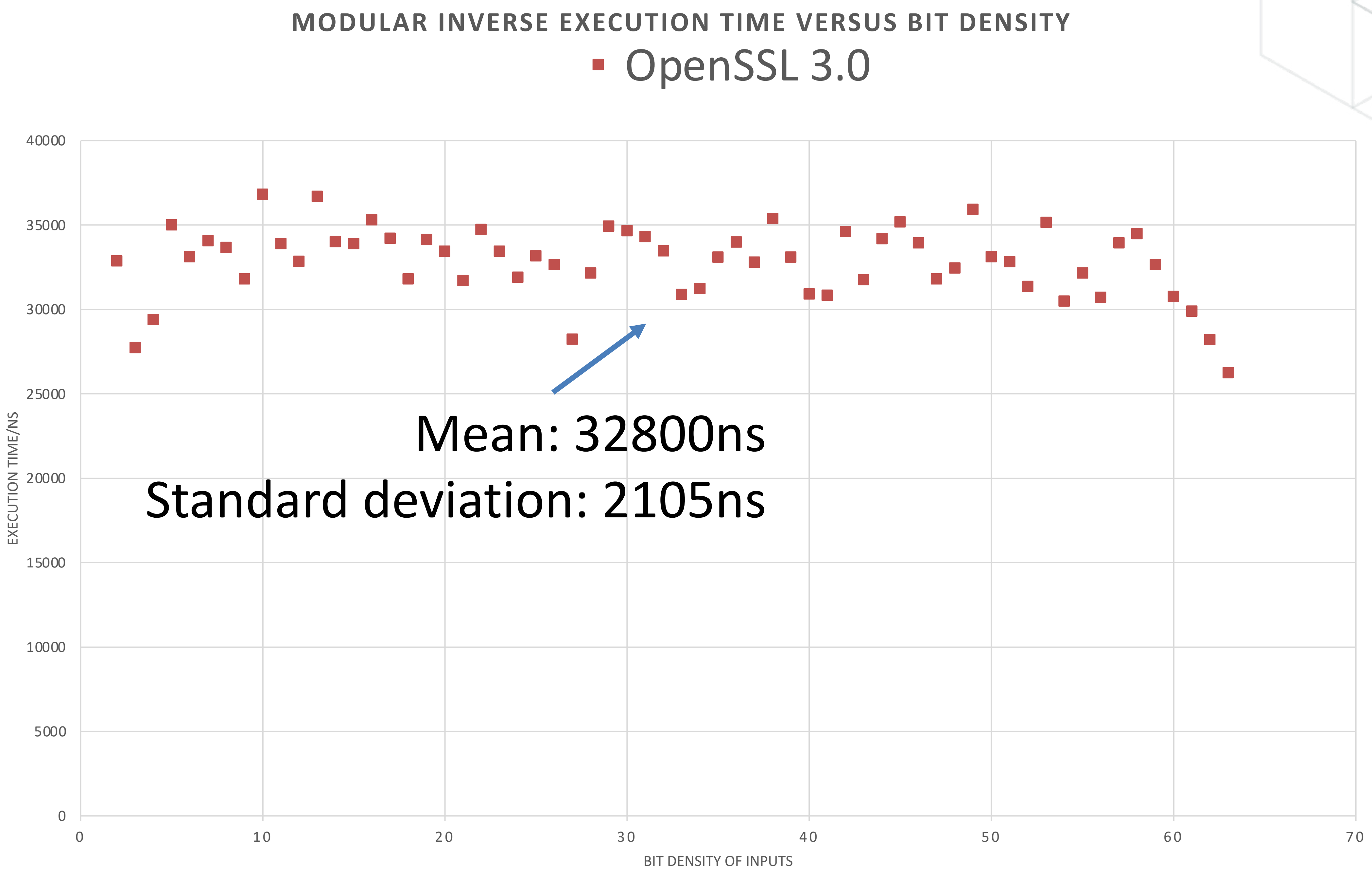
(Plus many lines deleted…)

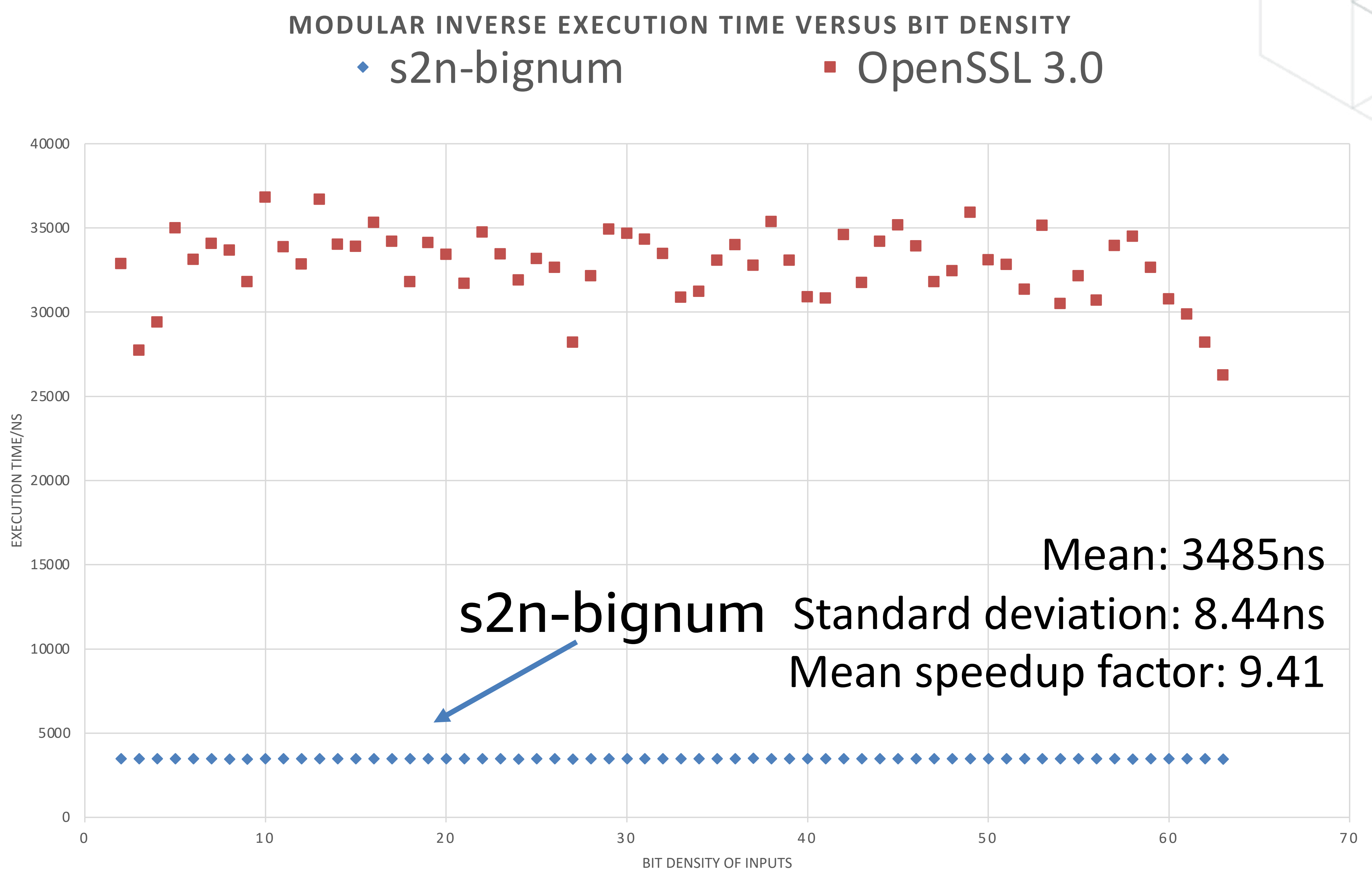
…then memory address z ends up containing the right answer!

s2n-bignum – performance?

- Performance of s2n-bignum is competitive or better than any other implementation, on most x86 and ARM64 micro-architectures…

- For example…Times for 384-bit modular inverse at bit densities 2–63
  - “bit density X” = “probability that randomly chosen input bit is a 1 is X/64”


MODULAR INVERSE EXECUTION TIME VERSUS BIT DENSITY
OpenSSL 3.0
Mean: 32800ns
Standard deviation: 2105ns
EXECUTION TIME/NS
BIT DENSITY OF INPUTS
40000
35000
30000
25000
20000
15000
10000
5000
0
10
20
30
40
50
60
70



MODULAR INVERSE EXECUTION TIME VERSUS BIT DENSITY
s2n-bignum
OpenSSL 3.0
EXECUTION TIME/NS
BIT DENSITY OF INPUTS
0
5000
10000
15000
20000
25000
30000
35000
40000
0
10
20
30
40
50
60
70
s2n-bignum
Mean: 3485ns
Standard deviation: 8.44ns
Mean speedup factor: 9.41

s2n-bignum – performance?

- How about RSA performance on Graviton-2 (64-bit ARM neoverse_n1 core)?
- Sign operations per second

Modulus size bits	Ops per second 1st January 2023	Ops per second 28th September 2023
2048	299	582
3072	96	139
4096	42	92

AR and Cryptography

Some Current Challenges…

Hybrid verification: what's the optimal mix of static and dynamic verification?

Will performance of PQ algorithms be a problem @ Cloud Scale?

PQC on low-speed, low-power "edge" devices?

AR and Cryptography

Some Current Challenges…

The devil-in-the-detail: where do we draw the line? How to cope with micro-architectural variation and defects?

Longevity…will notation X and/or tool Y still be viable in 20 years?

Can we achieve a “separation of concerns” between crypto mathematicians and software engineers? Very few people are world-class in *both* disciplines…

Takeaways

Automated Reasoning about and of the Cloud

Significant advances in reasoning about the correctness of our infrastructure and services.

Automated Reasoning in the Cloud

Got a big proof? Bring us your workloads!