

DNS権威サーバのクラウドサービス向けに 行われた攻撃および対策

JANOG51 Meeting in Fujiyoshida 2023/01/26

さくらインターネット株式会社 クラウド事業本部 SRE室 Masahiro Nagano (kazeburo)

Me

- 長野雅広(ながのまさひろ)
- @kazeburo Twitter/GitHub
- さくらインターネット株式会社
クラウド事業本部 SRE室 室長
- さくらインターネットの展示ブース@ふじさんホール2Fにいます

Me

- 2006年まで京都でスタートアップ、mixi、livedoor (現LINE)、mercariを経て2021年より現職
- これまでウェブアプリケーションの運用/SREをやってきました
- ISUCON1, 2, 9予選出題・ISUCON3, 4優勝
- JANOGは初参加になります



発売中

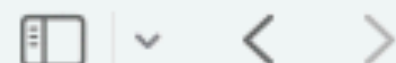
クラウド事業本部 SRE室

- 2022年7月に発足した新しい部署
- ミッション
 - クラウドサービスの信頼性を高めることにより、お客様や社会のDXをしっかりと支える
- ビジョン
 - 社内でのSREの実践を広め、お客様への価値提供を行う
 - さくらのサービスそのものの信頼性向上、それにより価値向上を目指す
 - さくら社員がEnabling SREとして、お客様・社外のサービスの信頼性向上に携わる

クラウド事業本部SRE室の取り組み

- Embedded SRE / Enabling SREとしての取り組み
 - クラウドサービスのチーム開発/運用体制作り
 - CI/CDなどDX(Developer Experience)向上の仕組みの構築
 - ポストモーテム導入
- SRE as a Service
 - 社内における Kubernetes 基盤構築
 - ログ/監視基盤の研究開発

JANOG39(2017年)の発表振り返り



www.janog.gr.jp/meeting/janog39/program/adns.html



Japan Network Operators' Group
日本ネットワーク・オペレーターズ・グループ

HOME

General Information ▾

Meetings ▾

Mailing List ▾

Archive ▾

Resource ▾

Sponsors

English Page



JANOG39は株式会社DMM.comラボのホストにより開催します。

開催概要

プログラム

LT投票結果

ストリーミング

出席登録

ホスト・協賛

ニュースレター

若者支援プログラム

スタッフ

現地情報

当日情報

DNS権威サーバ向けのDDoS攻撃対策をした話～さくらインターネット編～

概要

昨今、世界的にDNSサーバ宛のDDoS攻撃が頻発しています。

つい先日(10/21米国時間)にも、Dyn社のDNSサービスが攻撃を受け、多数の著名サイトに影響したことも記憶に新しいところ です。

弊社(さくらインターネット)でも、去る2016/8/29～9/2にかけて、お客様のゾーンを保持しているDNS権威サーバに対して断続的なDDoS攻撃を受け、ホスティングサービスを中心に大きな障害が発生しました。

残念ながら弊社DNS権威サーバは攻撃に強い構成ではありませんでした。

これを受け弊社では、半年程のスパンで「強いDNS」を作るべく、新クラスタの構築、IPアドレスのリナンバ、既存DDoSミティゲーションシステムの100Gアップグレード、L7ファイアウォールの導入、Anycastノード設置などの対策を順次実施しています。

本セッションでは、それらの取り組みを共有させていただき、参加者さんとの議論の中で、DNS権威サーバ向けの有効なDDoS対策についてアイデアをいただければと思っております。

事前公開資料

本セッションに関するDNSサービス向けのDDoS攻撃の参考情報として下記サイトをご紹介します。

JANOG39の発表振り返り

- 弊社DNSコンテンツサーバへのDDoS攻撃
- DDoSに耐えるためのバックボーンを含むインフラ構成の見直し
 - L7・DDoS Mitigation 装置の導入
 - 100Gトランジット導入
 - 上流でのDDoS対策の検討
- 今回の発表は「さくらのクラウド」のDNSサービスに行われた攻撃を扱う

さくらのクラウドの紹介および DNSアプリケーションの構成

さくらのクラウド

- 2011年のサービス開始から12年目に入りました
- 皆様のご支援のおかげです。改めて感謝申し上げます



さくらのクラウド

The screenshot displays the Sakura Cloud website interface. At the top, there is a navigation bar with the Sakura Cloud logo and various menu items like '機能・仕様' (Features/Specifications), '料金' (Pricing), and 'お問い合わせ' (Contact Us). Below the navigation bar, the main heading is 'さくらのクラウドの機能・仕様' (Sakura Cloud Features/Specifications). The content is organized into two main sections: '機能・仕様' (Features/Specifications) and '連携サービス' (Partner Services). The '機能・仕様' section is divided into eight categories, each with an icon, a brief description, and a '詳しく見る' (View Details) link. The '連携サービス' section is divided into three categories, each with an icon, a brief description, and a '詳しく見る' (View Details) link.

機能・仕様	連携サービス
サーバー/ディスク 正確なコストパフォーマンスと高い柔軟性を実現したクラウドサーバー。	ウェブアクセラレータ 手軽に使える高コストパフォーマンスCDNサービス。
ネットワーク 仮想データセンターを構築するような感覚で、柔軟なネットワーク設計・作成が可能。	さくらのセキュアモバイルコネクト 陸域網で通信速度制限なしの高セキュアなIoT/M2M向けSIM。
セキュリティ お客様のシステムを守る、VPN・ファイアウォール・WAF等の充実のセキュリティ機能。	オブジェクトストレージ 大容量データの保存に適した分散型クラウドストレージサービス。
負荷分散 高性能なロードバランサとDR対策に最適なGSLB(広域負荷分散)をご用意。	
ユーザーインターフェース シンプルで直感的なUIと「インフラの見える化」を実現したコントロールパネル。	
アクセスコントロール 2段階認証/マルチユーザー/操作権限設定により高セキュリティを実現。	
オプションサービス データベース・DNS・外部監視・SendGrid・NFSなどの充実のオプション機能。	
サービス間接続 専用サーバーやハウジングとの連携で柔軟なシステム構成を実現。	

- 東京と石狩リージョンで展開
- サーバ/ディスク・ネットワークなどIaaSを提供
- VPCルータ、データベースなどのアプライアンス
- 2拠点での冗長化を行うロードバランサ、GSLB、DNSアプライアンス
- オートスケール

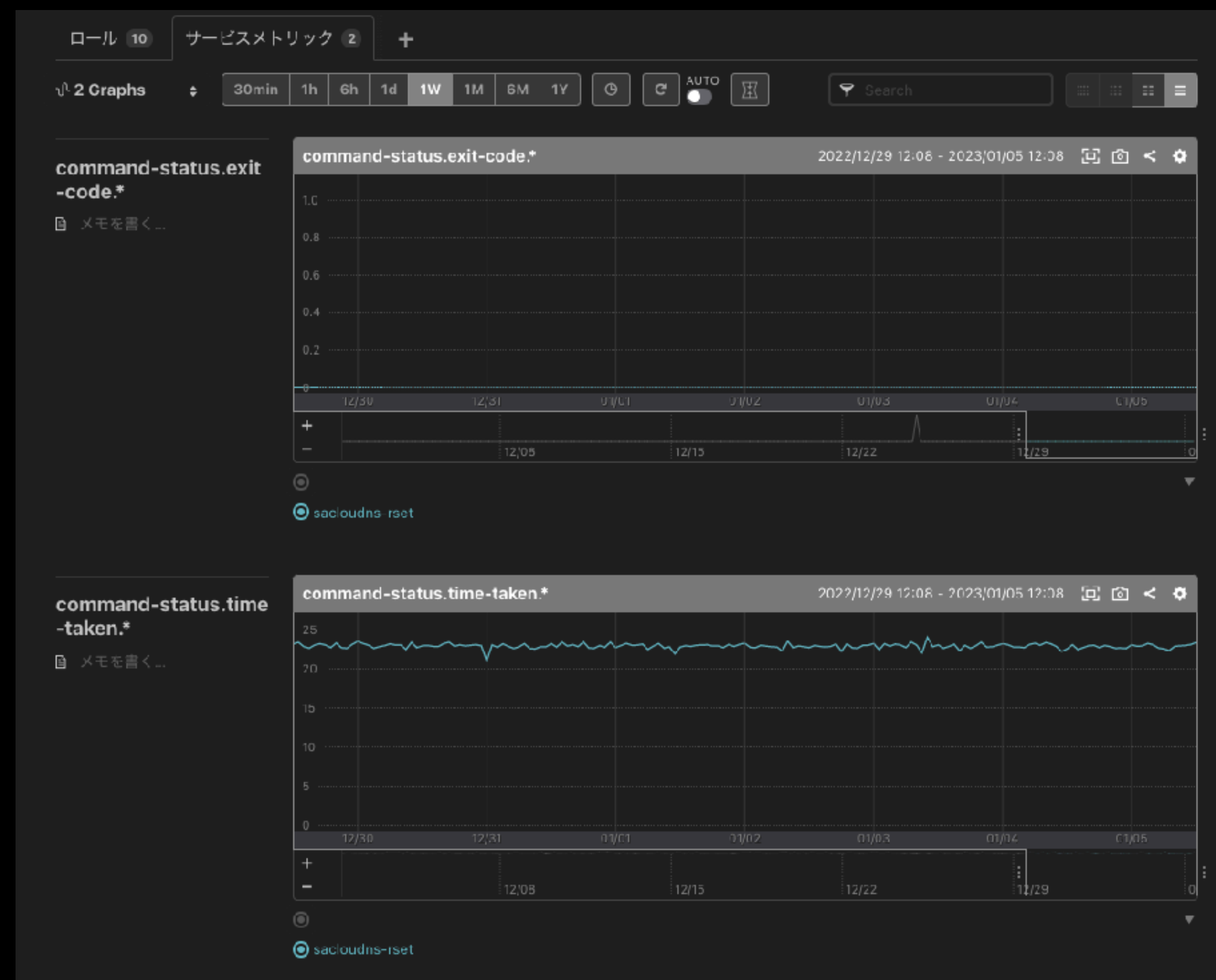
DNSアプライアンス

- 権威DNSサーバのクラウドサービス
- お客様が所有するドメインのゾーン情報などをコントロールパネルやAPIで管理
- 一般的なレコードタイプに加え、ALIASやHTTPS RRに対応



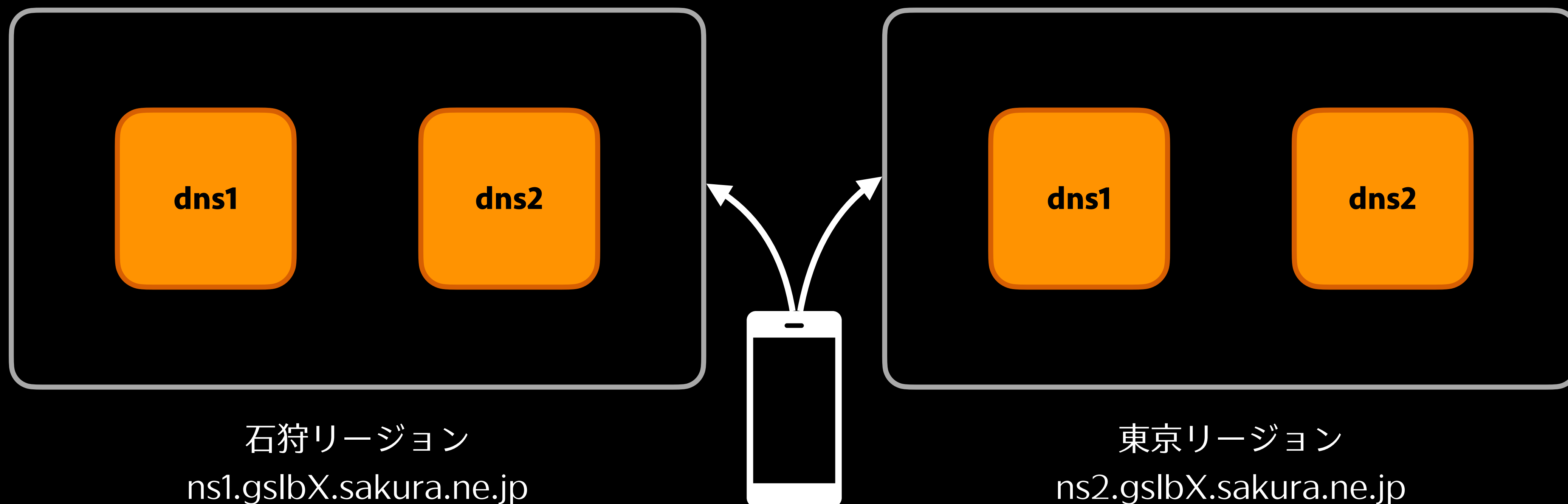
API操作からDNS浸透反映までの速度

- Mackerelをつかって見える化する
- API呼び出し含み23秒前後で反映
- SRE的取り組みのひとつ



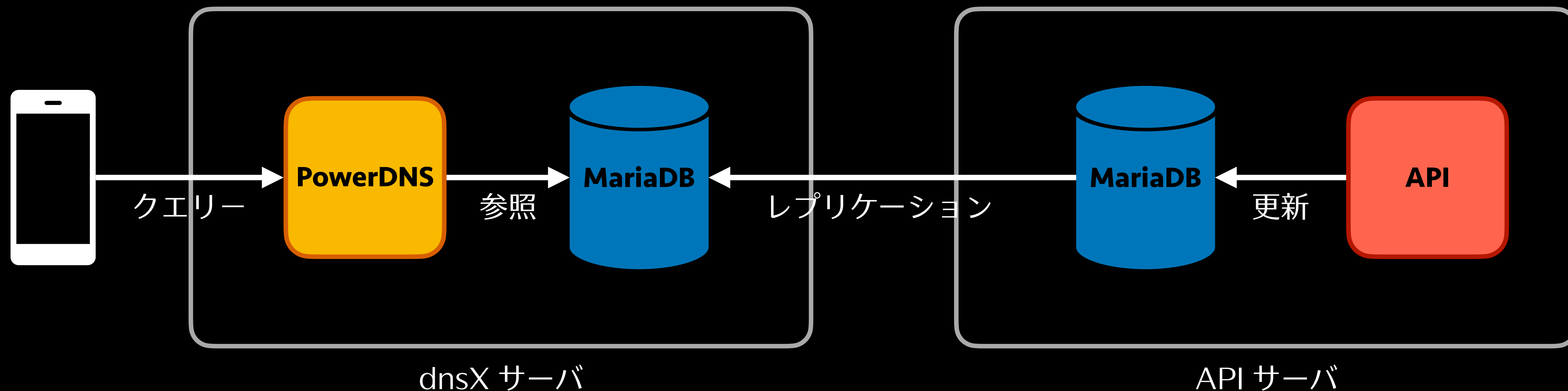
DNSアプリケーションの構成

- DNSは石狩と東京でリージョン分散
- それぞれのリージョンでも複数台のサーバで冗長化



DNSアプリケーションの構成

- 権威DNSサーバには PowerDNS Authoritative Server を採用
 - Backendとして RDBMS (MariaDB) を使用



DNSサーバへの攻撃

なぜDNSサーバが狙われるのか

- DNSが動作しなければ一般ユーザはインターネットが利用できないとの同じ
 - DNSサーバの運用主体への脅迫・抗議
 - 特定のWebサイトへのアクセスを不能にさせる
 - 正当なルートを改ざんし、不正サイトに誘導(フィッシングなど)

DNSサーバへの攻撃の分類

- DoS/DDoSによるサービス妨害
 - DNSフラッド攻撃
 - DNS水責め攻撃
- DNSの改ざん
 - キャッシュポイズニング

「DNS水責め攻撃」とは

- ランダムサブドメイン攻撃 (Pseudo-Random Subdomain Attack) と呼ばれることも
- 2014年に初めて観測 (<https://cybersecurity-jp.com/column/34745>)
- 2014年～2016年に攻撃や議論が多く行われている

「DNS水責め攻撃」とは

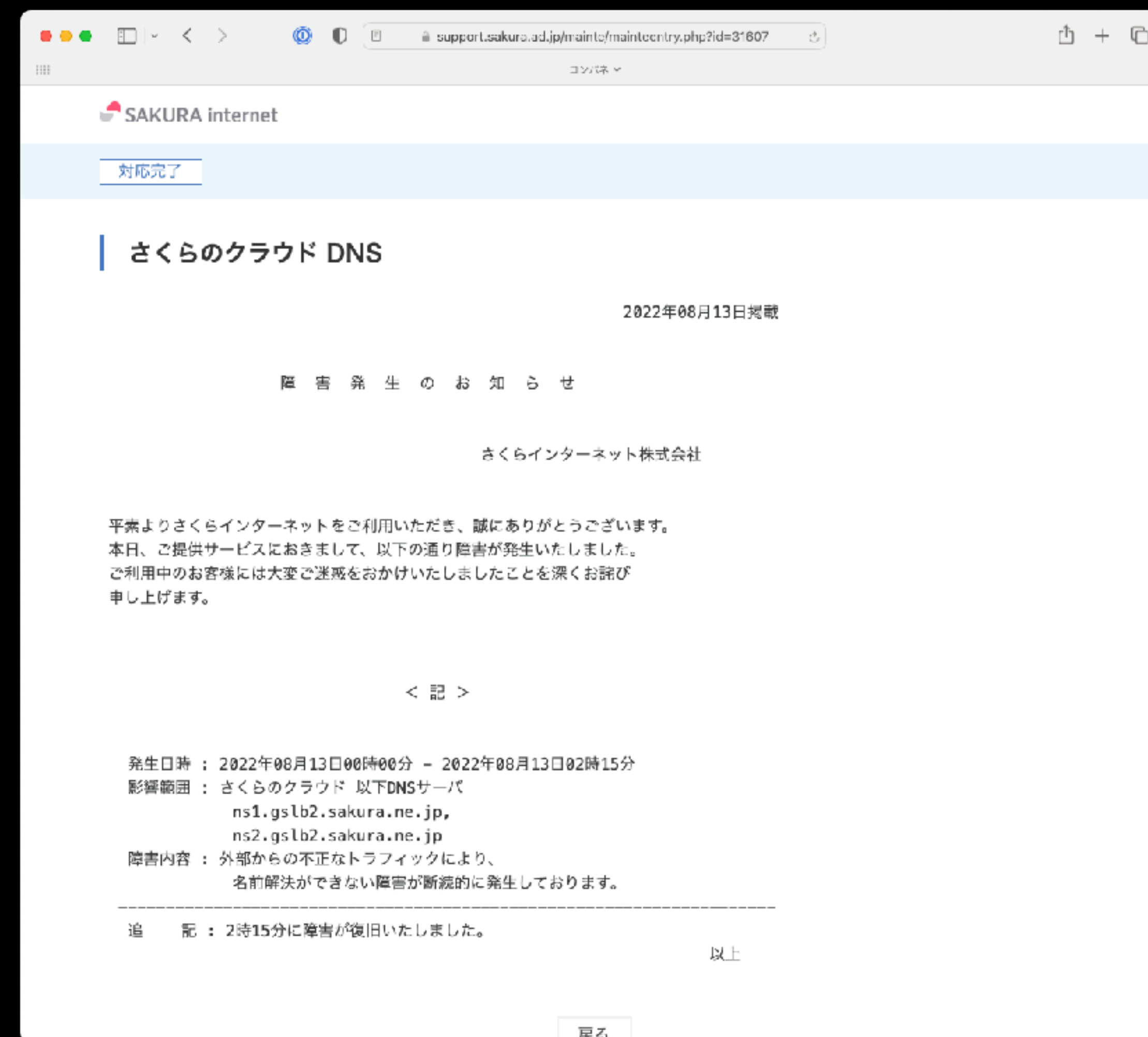
- 攻撃対象に大量のランダムなサブドメインを問い合わせさせてDNSの機能停止、機能低下を狙う攻撃
- 攻撃者はオープンリゾルバに対して、大量のランダムサブドメインの問い合わせを発生させる
- DNSキャッシュサーバにはネガティブキャッシュまで含めて、キャッシュが存在しない
- その結果、権威DNSサーバに問い合わせが発生し、DoSとなる

「DNS水責め攻撃」とは

- フラッド攻撃のような高帯域とはならない
- DNSクエリとして正しいパケット・動作であり防ぐのが難しい

さくらのクラウド DNSアプライアンスへの攻撃

- 2022年8月に発生。断続的に攻撃が続いている
- 数度に渡りサービスへの影響



The screenshot shows a web browser window with the URL `support.sakura.ad.jp/maint/maintentry.php?id=31607`. The page header includes the SAKURA internet logo and a status indicator "対応完了" (Completed). The main heading is "さくらのクラウド DNS" (Sakura Cloud DNS), with a sub-heading "2022年08月13日掲載" (Posted on August 13, 2022). The content is titled "障害発生のお知らせ" (Notice of Service Disruption) from "さくらインターネット株式会社" (Sakura Internet Co., Ltd.). The text states: "平素よりさくらインターネットをご利用いただき、誠にありがとうございます。本日、ご提供サービスにおきまして、以下の通り障害が発生いたしました。ご利用中のお客様には大変ご迷惑をおかけいたしましたことを深くお詫び申し上げます。" (Thank you for your continued use of Sakura Internet. Today, a service disruption occurred as follows. We deeply apologize for the inconvenience caused to our customers.) Below this, a "＜ 記 ＞" (Note) section provides details: "発生日時：2022年08月13日00時00分 - 2022年08月13日02時15分" (Occurrence time: 2022/08/13 00:00 - 2022/08/13 02:15), "影響範囲：さくらのクラウド 以下DNSサーバ" (Impact: Sakura Cloud, below DNS servers), "ns1.gslb2.sakura.ne.jp, ns2.gslb2.sakura.ne.jp" (Affected servers), and "障害内容：外部からの不正なトラフィックにより、名前解決ができない障害が断続的に発生しております。" (Disruption content: Due to malicious traffic from outside, intermittent disruptions in name resolution have occurred). A "追 記" (Update) at the bottom states: "2時15分に障害が復旧いたしました。" (The disruption was resolved at 2:15). The page ends with "以上" (End) and a "戻る" (Back) button.

断続的に続く攻撃



実際の攻撃の記録(1分間あたりのクエリ数)



12/15から12/16まで1日近く、900万クエリ/分の攻撃が続いた

実際の攻撃の記録(tcpdump)

```
07:25:11.719035 IP 209.216.160.2.50051 > 133.242.64.100.53: 43104 A? meetmodeling.example.com. (50)
07:25:11.719057 IP 205.171.30.238.44916 > 133.242.64.100.53: 64321% [1au] A? _.modeling.example.com. (71)
07:25:11.719069 IP 172.70.109.31.63292 > 133.242.64.100.53: 40380 [1au] A? osaExpe1-pLatINUM.exAmPLe.c0m. (66)
07:25:11.719071 IP 3.139.136.204.44597 > 133.242.64.100.53: 32383% [1au] A? webdirect.foster.example.com. (65)
07:25:11.719113 IP 18.188.77.103.42513 > 133.242.64.100.53: 14853 [1au] A? note-modeling.example.com. (62)
07:25:11.719132 IP 172.70.33.19.27971 > 133.242.64.100.53: 35379 [1au] A? indian-awarded.example.com. (63)
07:25:11.719150 IP 172.70.33.19.27971 > 133.242.64.100.53: 35379 [1au] A? matchfiling.example.com. (49)
07:25:11.719168 IP 172.70.33.19.27971 > 133.242.64.100.53: 35379 [1au] A? matchfiling.example.com. (49)
07:25:11.719186 IP 172.70.33.19.27971 > 133.242.64.100.53: 35379 [1au] A? matchfiling.example.com. (49)
07:25:11.719204 IP 172.70.33.19.27971 > 133.242.64.100.53: 35379 [1au] A? matchfiling.example.com. (49)
07:25:11.719222 IP 172.70.33.19.27971 > 133.242.64.100.53: 35379 [1au] A? matchfiling.example.com. (49)
07:25:11.719240 IP 172.70.33.19.27971 > 133.242.64.100.53: 35379 [1au] A? matchfiling.example.com. (49)
07:25:11.719258 IP 172.70.33.19.27971 > 133.242.64.100.53: 35379 [1au] A? matchfiling.example.com. (49)
07:25:11.719275 IP 96.114.53.69.53157 > 133.242.64.100.53: 5679 [1au] A? gitcn-awarded.example.com. (62)
07:25:11.719312 IP 172.70.229.30.59530 > 133.242.64.100.53: 45890 [1au] A? ipafoster.example.com. (58)
07:25:11.719336 IP 172.217.46.78.59507 > 133.242.64.100.53: 60186% [1au] A? testcloud-modeling.example.com. (67)
07:25:11.719351 IP 69.47.193.166.52891 > 133.242.64.100.53: 238 [1au] A? bfmpassing.example.com. (59)
07:25:11.719353 IP 34.218.119.91.26001 > 133.242.64.100.53: 31511% [1au] A? signal-modeling.example.com. (64)
07:25:11.719365 IP 34.218.119.91.13381 > 133.242.64.100.53: 4210% [1au] A? pairfiling.example.com. (59)
```

- ランダムな文字列、単語の組み合わせ
- 大文字・小文字まざり(Google Public DNS仕様)
- ラベル数が増えることも

実際の攻撃の記録(攻撃元)

```
# zgrep -i example.com tcpdump_20221216-0725.txt.gz | awk '{print $3}' | awk -F. '{print $1"."$2".x.x"}' | sort  
| uniq -c | sort -hr | head -20  
159123 172.253.x.x # public DNS提供者A  
96013 74.125.x.x # public DNS提供者A  
63560 172.70.x.x # public DNS提供者B  
48554 172.71.x.x # public DNS提供者B  
44872 18.217.x.x # クラウド大手C  
42478 3.139.x.x # クラウド大手C  
42057 3.18.x.x # クラウド大手C  
39979 3.142.x.x # クラウド大手C  
29020 3.228.x.x # クラウド大手C  
28547 8.0.x.x  
27688 172.217.x.x  
27478 44.192.x.x  
23852 172.68.x.x  
22859 173.194.x.x  
19080 165.225.x.x  
17485 192.221.x.x  
17328 172.69.x.x
```

- Public DNS提供者A, Bが多い
 - オープンリゾルバを踏み台にしている
- 日によって傾向が異なることもある
 - 米国以外、ロシアなどのIPが混じることもある

DNSアプリケーションが攻撃の影響を受けやすい理由

- 多くのレコードを管理しやすくするためRDBMS (MariaDB) backendを利用
- 水責め攻撃ではキャッシュは有効に働かず、都度バックエンドに対して「SQL」が発行され、比較的重い処理となる
- CPU負荷による応答が遅延、PowerDNSのダウン(後述)

水責め攻撃への対応と対策

攻撃検知から実際の対策(初回)

- CPU負荷があがっての名前解決遅延
- 冗長化のためのVRRPでの切り替えおよび、切り戻りでも名前解決できない時間が発生
 - スタンバイ側を停止する対応
- 夜間であり収束するまで待つ
 - 長時間にわたり、影響

攻撃検知から実際の対策(二度目以降)

- iptables による対策(次のページ)
- サーバのスケールアップ
 - DNSサーバもさくらのクラウドのIaaSの上に展開されているためスケールアップは短時間で可能
- DDoS Mitigation 装置(JANOG39で紹介)の導入
- PowerDNS、MariaDBのチューニング

iptablesでの対策

```
# *.example.com の問い合わせを落とす
```

```
iptables -I INPUT 14 -i eth0 -p udp --dport 53 -m string --hex-string "|  
076578616d706c6503636f6d000001|" --algo bm --from 41 --to 512 -j DROP -m comment --  
comment "*.example.com:a:udp"
```

```
iptables -I INPUT 14 -i eth0 -p tcp --dport 53 -m string --hex-string "|  
076578616d706c6503636f6d000001|" --algo bm --from 67 --to 512 -j DROP -m comment --  
comment "*.example.com:a:tcp"
```

```
# www.example.com の問い合わせは許可する
```

```
iptables -I INPUT 14 -i eth0 -p udp --dport 53 -m string --hex-string "|  
777777076578616d706c6503636f6d000001|" --algo bm --from 41 --to 512 -j ACCEPT -m comment --  
comment "www.example.com:a:udp"
```

```
iptables -I INPUT 14 -i eth0 -p tcp --dport 53 -m string --hex-string "|  
777777076578616d706c6503636f6d000001|" --algo bm --from 67 --to 512 -j ACCEPT -m comment --  
comment "www.example.com:a:tcp"
```

PowerDNSチューニング(1)

- iptables、DDoS Mitigation 装置を導入したことで影響
 - DNSサーバの手前でパケットをDropすることで、サーバ側にTCP接続が残ってしまう現象の発生
 - TCP接続の最大数を超過してしまい、TCPでの名前解決ができなくなる
 - max open filesの緩和とともに、PowerDNSのTCP設定をチューニング

```
tcp-idle-timeout=1 // 早期に切断する
```

```
max-tcp-connections=1500
```


PowerDNSチューニング(2)

- PowerDNSはバックエンドへの問い合わせが貯まると自動でダウン
 - max-queue-length という設定。デフォルト 5,000
 - PowerDNSがダウンし、systemdによって再起動されるが、その間は接続不能となる
 - max-queue-length を増やすことで落ちにくくはなるがレイテンシは悪化する

対策の改善へ

- iptables / DDoS Mitigation装置での対策の問題点
 - iptablesでは大文字小文字混じりのクエリは扱えない
 - 大規模な攻撃では影響を受ける可能性
 - DDoS Mitigation装置では攻撃検知するとゾーン丸ごとレートリミットがかかる仕様
 - お客様影響が避けられない
 - 攻撃者の狙いを回避できているか

対策の改善へ

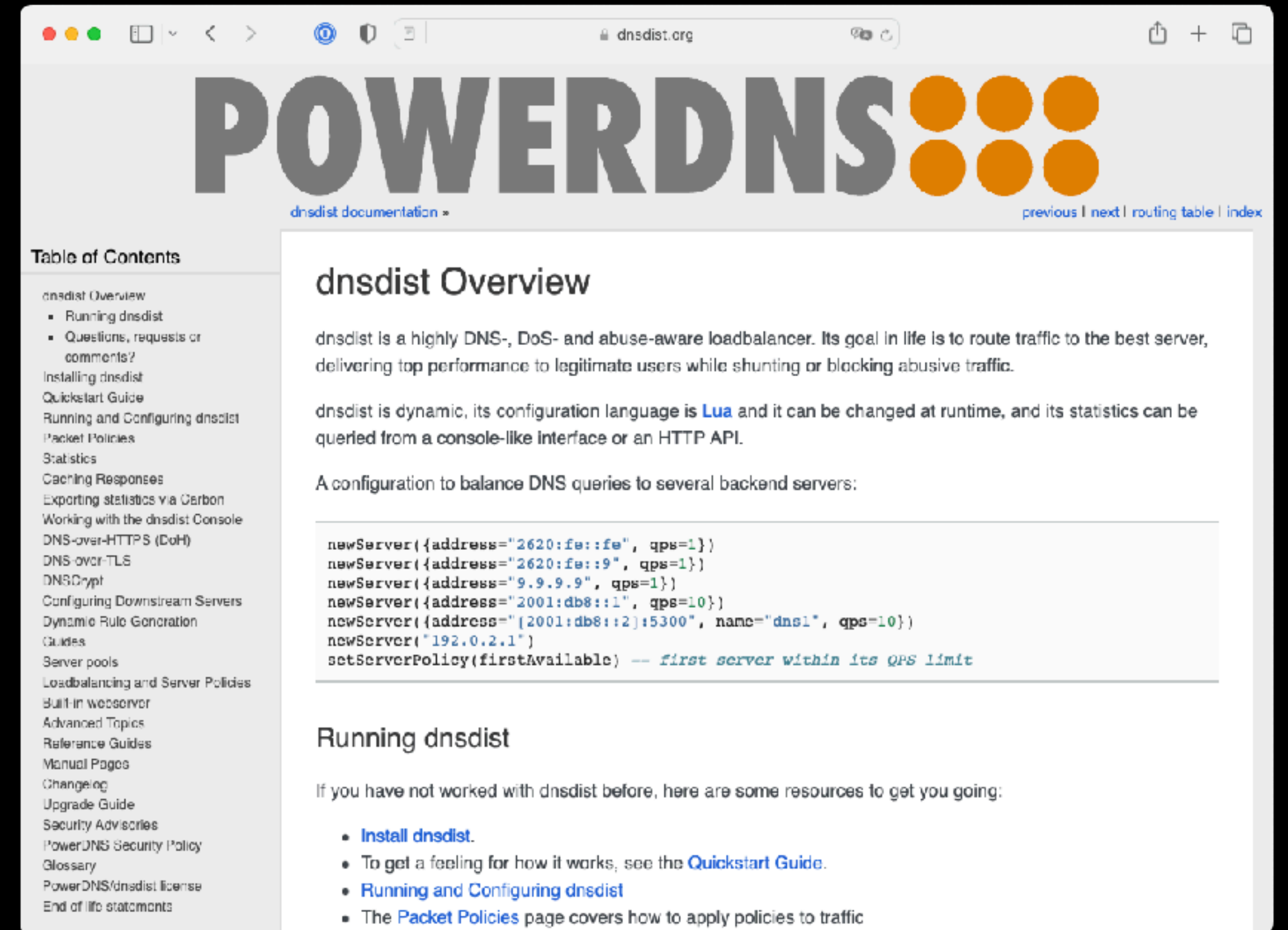
- お客様操作に影響
 - お客様にてレコードの追加をしてもiptablesでブロックされ名前解決不可
 - カスタマーサポートから連絡も行っていた

対策の改善へ

- サーバ側でクエリの中身を見て細かく判断
 - dnsmdistの導入
- モニタリング改善

dnsmist (https://dnsmist.org/)

- PowerDNSの開発元がOSSとしてリリースしているDNSのプロキシサーバ
- dnsmist is a highly DNS-, DoS- and abuse-aware loadbalancer. Its goal in life is to route traffic to the best server, delivering top performance to legitimate users while shunting or blocking abusive traffic.



The screenshot shows the dnsmist documentation website. The main heading is "POWERDNS" with a logo of four orange circles. Below it, the text reads "dnsmist documentation" and "previous | next | routing table | index". The "Table of Contents" on the left lists various topics, with "dnsmist Overview" selected. The main content area is titled "dnsmist Overview" and contains the following text:

dnsmist is a highly DNS-, DoS- and abuse-aware loadbalancer. Its goal in life is to route traffic to the best server, delivering top performance to legitimate users while shunting or blocking abusive traffic.

dnsmist is dynamic, its configuration language is [Lua](#) and it can be changed at runtime, and its statistics can be queried from a console-like interface or an HTTP API.

A configuration to balance DNS queries to several backend servers:

```
newServer({address="2620:fe::fe", qps=1})
newServer({address="2620:fe::9", qps=1})
newServer({address="9.9.9.9", qps=1})
newServer({address="2001:db8::1", qps=10})
newServer({address="[2001:db8::2]:5300", name="dns1", qps=10})
newServer("192.0.2.1")
setServerPolicy(firstAvailable) -- first server within its QPS limit
```

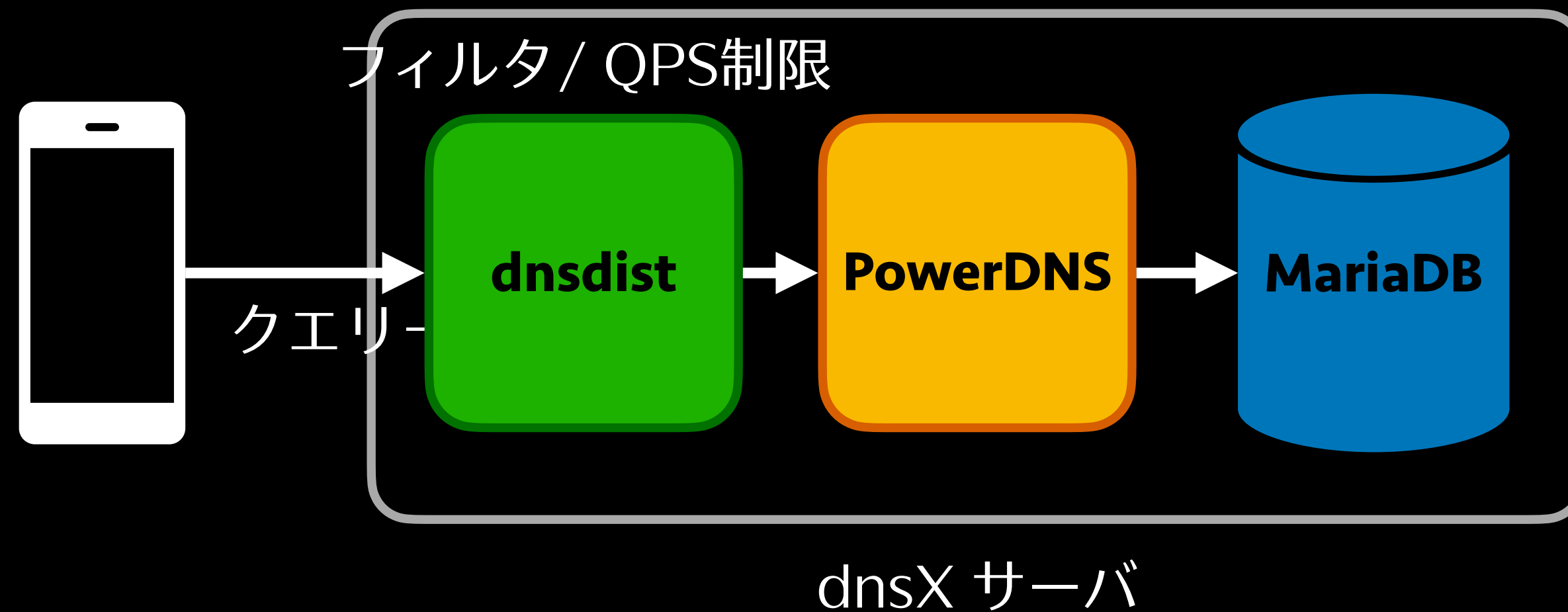
Running dnsmist

If you have not worked with dnsmist before, here are some resources to get you going:

- [Install dnsmist.](#)
- To get a feeling for how it works, see the [Quickstart Guide](#).
- [Running and Configuring dnsmist](#)
- The [Packet Policies](#) page covers how to apply policies to traffic

dnsmdistを含む構成

- 既存のDNSサーバ上に dnsmdist を導入。PowerDNSを別ポートで動かす
- PowerDNSへのクエリのフィルタ、QPS制限を行う



dnsmdist設定

```
addLocal("0.0.0.0:53", {reusePort=true})
addLocal("0.0.0.0:53", {reusePort=true})

newServer({address="127.0.0.1:1053",name="backend1"})
newServer({address="127.0.0.1:1053",name="backend2"})
setServerPolicy(roundrobin)

domain1 = newSuffixMatchNode()
domain1:add(newDNSName("example.com.))
addAction(
  AndRule({
    SuffixMatchNodeRule(domain1),
    OrRule({QTypeRule(DNSQType.A),QTypeRule(DNSQType.AAAA)}),
    NotRule(QNameRule("example.com.)),
    NotRule(QNameRule("www.example.com.)),
    MaxQPSIPRule(3,16)
  }),
  DropAction()
)
```

- 上流はローカルホストの1053ポート
- ネイキッドドメイン(Zone Apex)
www以外にQPS制限
- /16 で 3QPS以上はDropする
- 正しいサブドメインは影響受けない

dnsdistベンチマーク

- prsd-benchという負荷ツールをGo言語で作って導入前に検証
- 検証環境にてdnsdistを「2QPSを超えたらRefuseを返す」に設定
 - 16万qps(960万クエリ/分)は捌けることを確認(CPU 4コア)

```
# GOGC=500 ./prsd-bench -P 53 -H 192.168.10.50 --max-workers 200 --max-length 8 -zone example.com
2023-01-06 11:17:34.853910735 +0900 JST m=+10.001231332 resolved: 2.100000 query/sec, refused 161820.500000
query/sec, failed 0.000000 query/sec
2023-01-06 11:17:44.856551706 +0900 JST m=+20.003872303 resolved: 2.000000 query/sec, refused 164345.000000
query/sec, failed 0.000000 query/sec
2023-01-06 11:17:54.853914469 +0900 JST m=+30.001235065 resolved: 2.000000 query/sec, refused 162952.400000
query/sec, failed 0.000000 query/sec
```


モニタリングと対応の改善

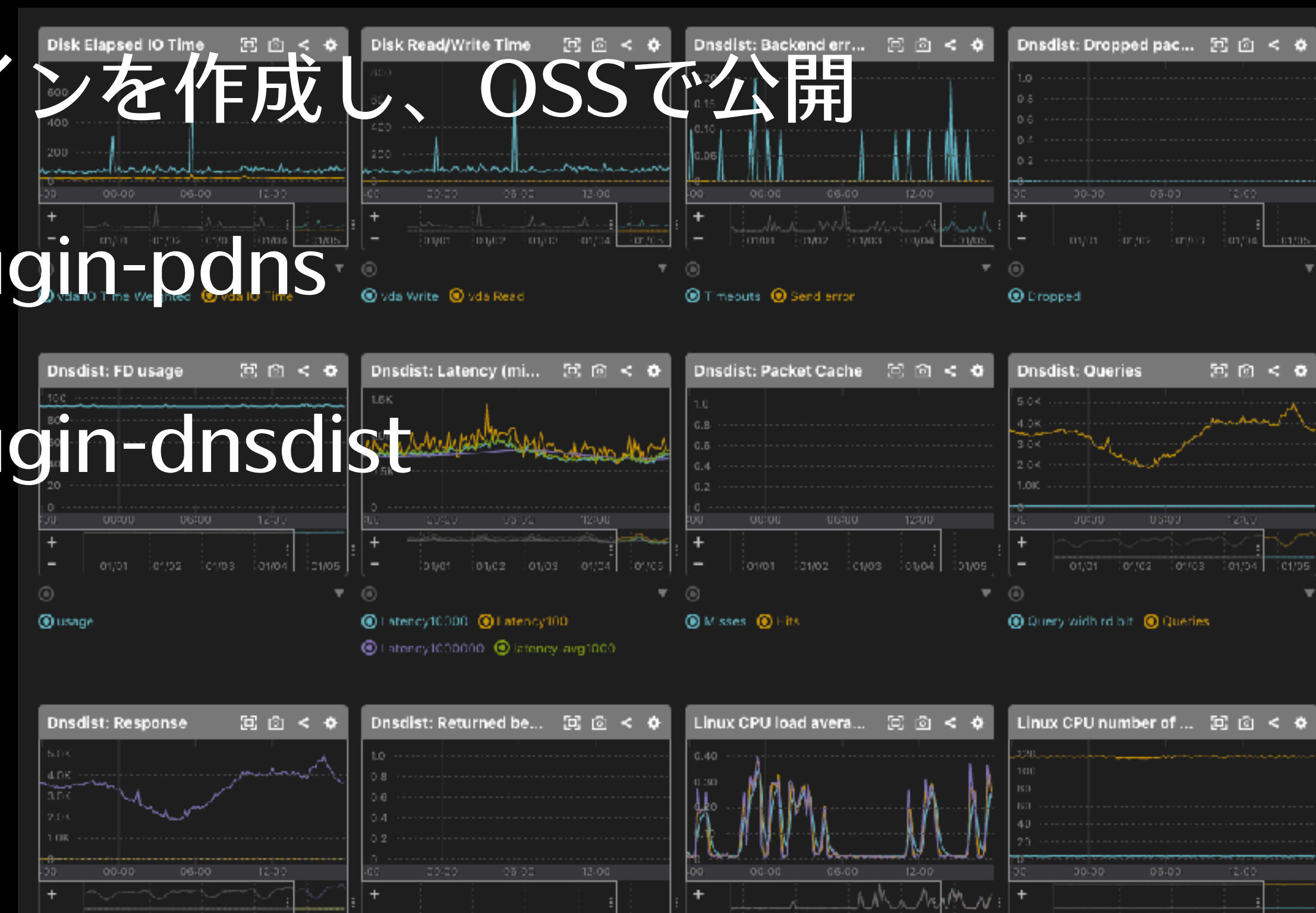
- Mackerelを利用してサーバのメトリクスを収集

- PowerDNSおよびdnsmdistの監視プラグインを作成し、OSSで公開

- github.com/kazeburo/mackerel-plugin-pdns

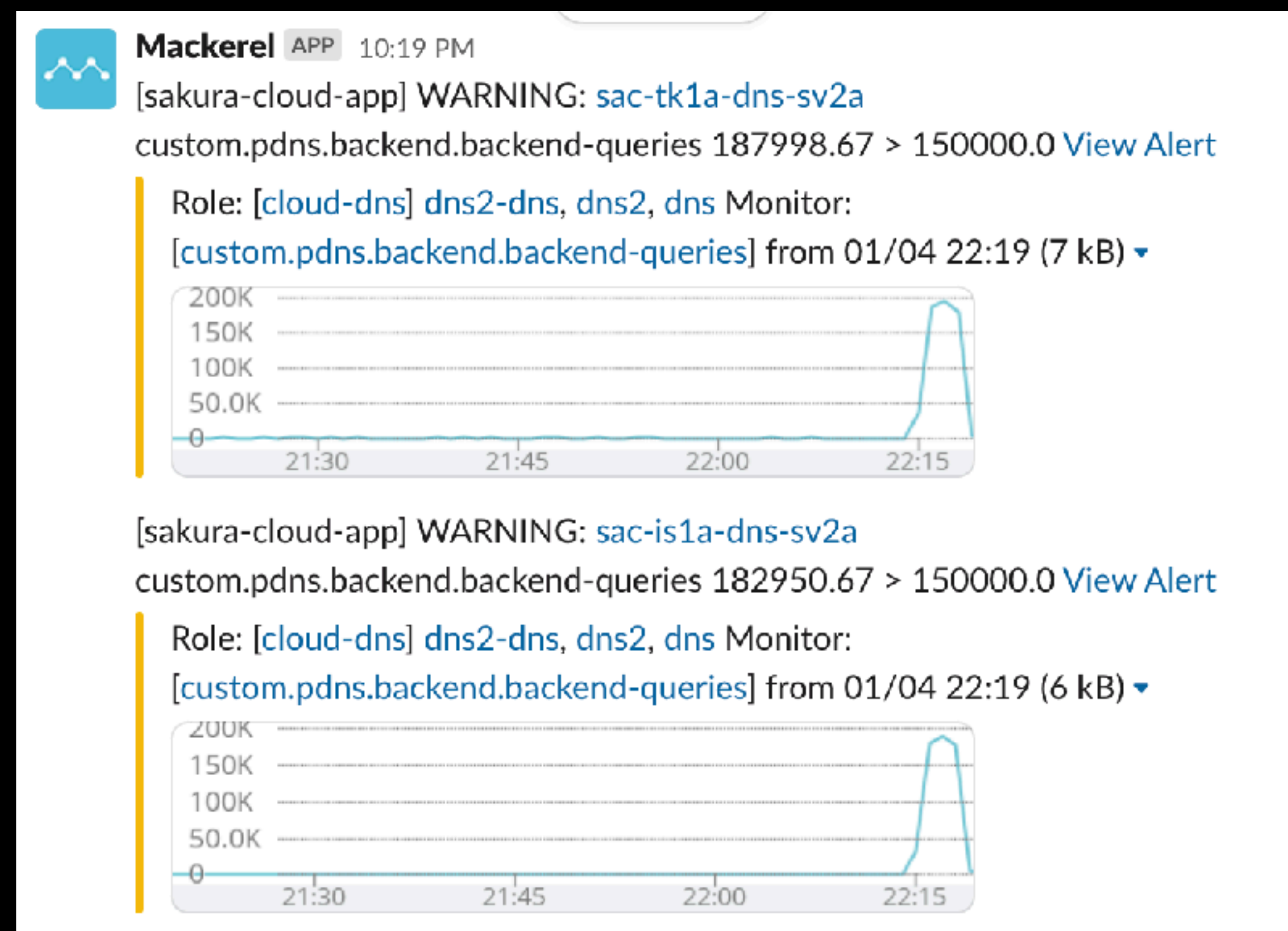
- github.com/kazeburo/mackerel-plugin-dnsmdist

- MariaDB含め様々なメトリックを収集



モニタリングと対応の改善

- 攻撃検知時はSlackへ通知
 - 加えてサーバ上にtcpdumpにて自動でログ記録
- 新たなゾーンへの攻撃時には dnsmist の設定を作り、Ansibleで投入する手順を作成、共有



「DNS水責め攻撃」対策の難しさ

攻撃対策の難しさ(1)

- パブリックDNS(オープンリゾルバ)からの大量アクセス
 - 他ゾーンの名前解決に影響があり、単純なQPS制限の適用が困難
- ゾーン単位でのQPS制限
 - サービス拒否攻撃に繋がる。攻撃者の狙いを回避できない

攻撃対策の難しさ(2)

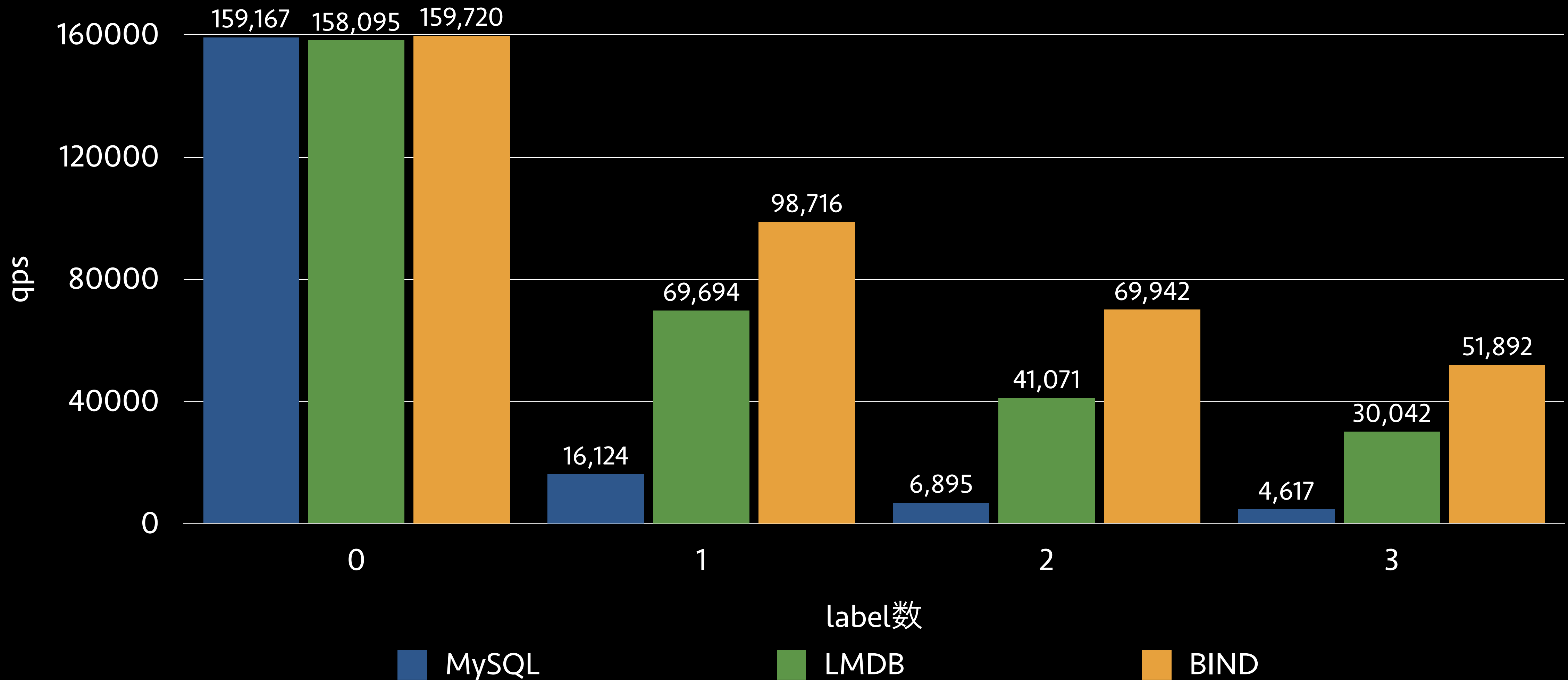
- ホワイトリストの規模が大きくなることで負荷が増大
- ワイルドカードの扱いが困難

今後の対策案

- MySQL(MariaDB) backendをやめる
 - LMDB、BIND形式だと7-8倍以上の性能向上
 - ゾーンまるごとキャッシュ
 - PowerDNS 4.8系のマイルストーンに対策は上がっているが...
- 負荷分散・オートスケール
- 攻撃に対する対応・SLAの明記
- 反映遅延の許容範囲の緩和(マニュアルには1分と記載)

PowerDNS Backend毎のベンチマーク

Backend / label数ベンチマーク



まとめ

まとめ

- さくらのクラウド DNSアプリケーションの構成
- 実際に発生した水責め攻撃および、対策
- DNS水責め攻撃対策の難しさ

議論したいこと

- PowerDNSやdnsmdistの運用ノウハウ
- DNS水責め攻撃の対策してますか？
 - 影響やとっている対策など
 - 返すべきレスポンスについて
- DNSサーバからみて攻撃元となるオープンリゾルバでの対策の可能性について

ご清聴ありがとうございました