

# 自社サービスの IPv6 対応、 最初の一歩が長かった

2023/01/26

JANOG51 Meeting

株式会社ビットフォレスト 野村 健太郎

# 目次

1. はじめに
2. IPv6 調査・検証の道のり
3. ハマったこと・困ったこと・よかったこと
4. 議論・質疑応答

はじめに

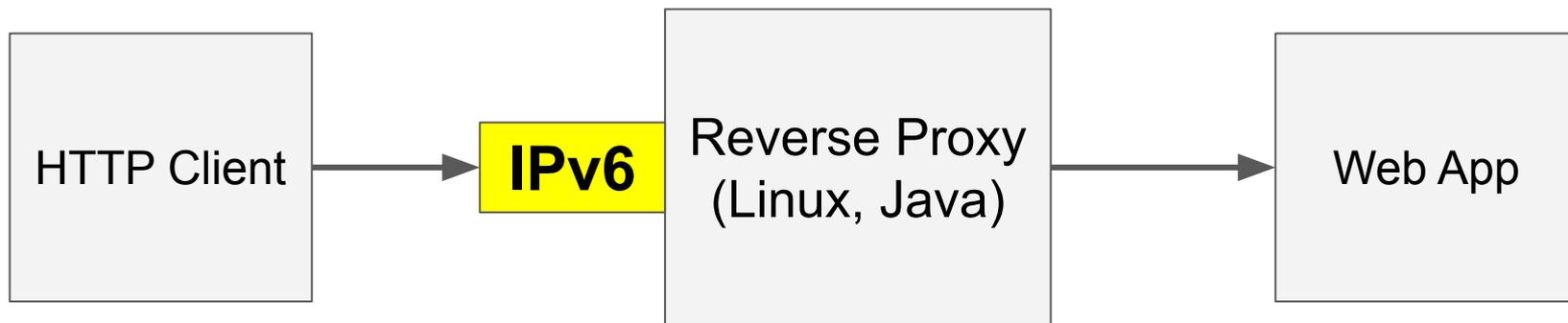
# 自己紹介

- 名前: 野村 健太郎
- 所属: 株式会社ビットフォレスト
  - Web セキュリティサービスの会社
- 経歴
  - ID 管理, SSO: LDAP, SAML, OAuth 2.0, OpenID Connect
  - Linux & OpenSource : OpenLDAP, OpenAM, Shibboleth
  - Infrastructure: AWS、その他 IaaS
  - DevOps: Docker, Ansible
- 趣味
  - バイクツーリング、バイクレース観戦、鈴菌感染者
- JANOG 初参加・初登壇

# 自社 Web サービス IPv6 対応

- 対象

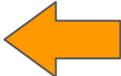
- リバースプロキシ
- ネットワークプログラミングあり(Java)



# IPv6 対応の工程

1. IPv6 技術調査
2. IPv6 対応可否検討
3. IPv6 対応実施
4. IPv6 サービス開始

# IPv6 対応の工程

1. IPv6 技術調査  今日お話しする「**最初の一歩**」の部分
2. IPv6 対応可否検討
3. ~~IPv6 対応実施~~
4. ~~IPv6 サービス開始~~

まだ IPv6 の需要があまりない、

対応のための工数がかかる、

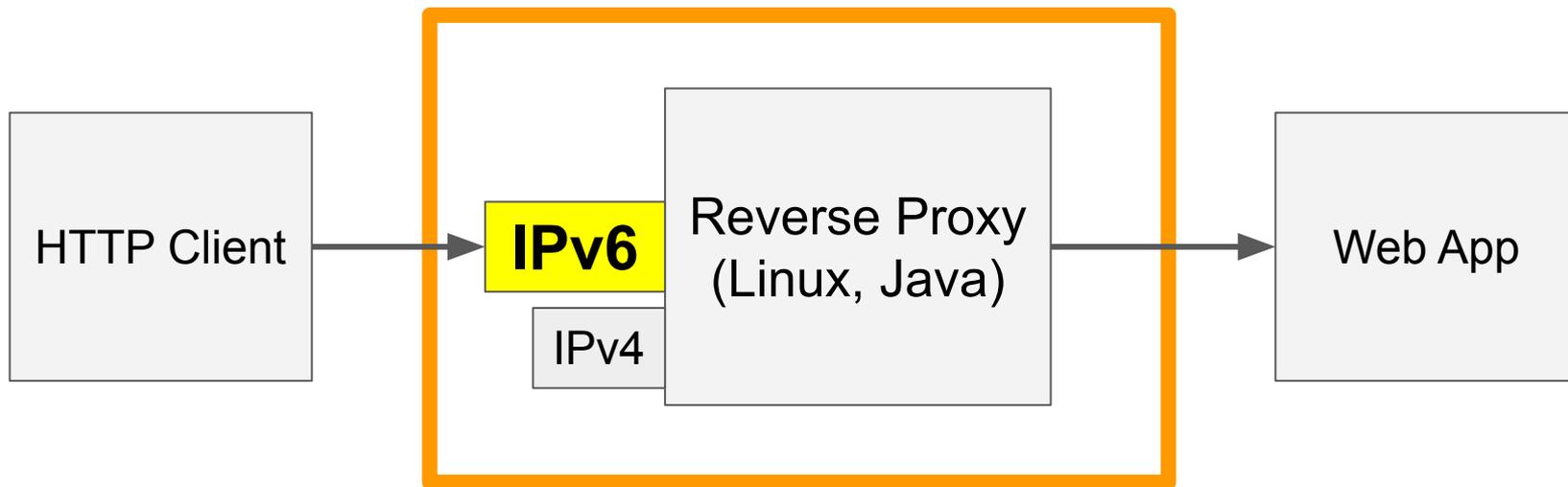
→IPv6 対応は保留に...

# 今回の発表に至る経緯

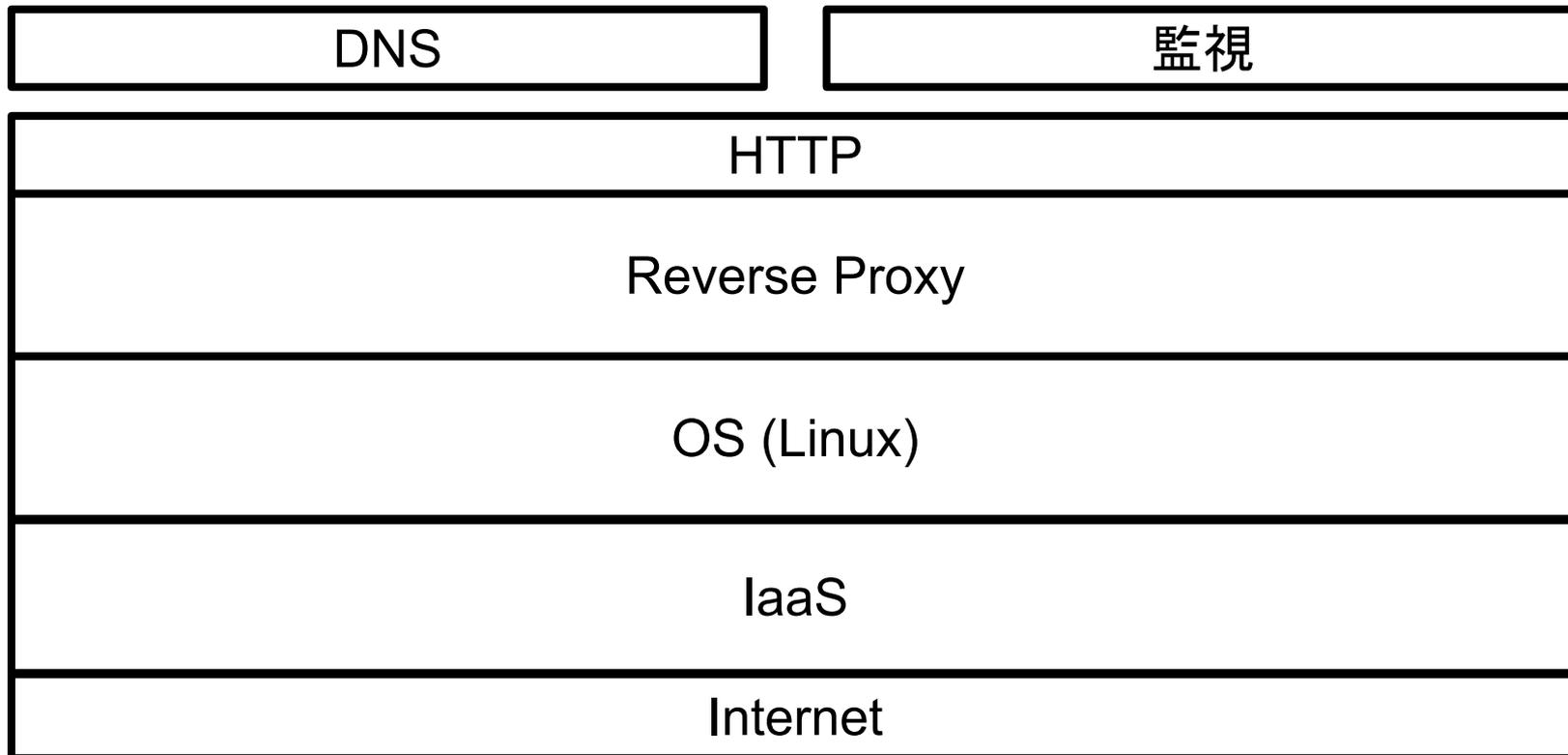
- 自社 Web サービスの IPv6 対応を検討することになった
- IPv6 経験者はおらず、ゼロからのスタート
- 手探りで調査・検証を進め、IPv6 対応の道筋をつけることはできた。時間がかかった(約半年)
- 苦労した点多々あり、この経験を共有したいなと思っていた
- JANOG51 のテーマは「**動き出す、一歩**」。「これだ！」と思って意を決して応募したら採用された！
- これから IPv6 を学ぶ皆さんの参考になれば嬉しい

# IPv6 調査・検証の道のり

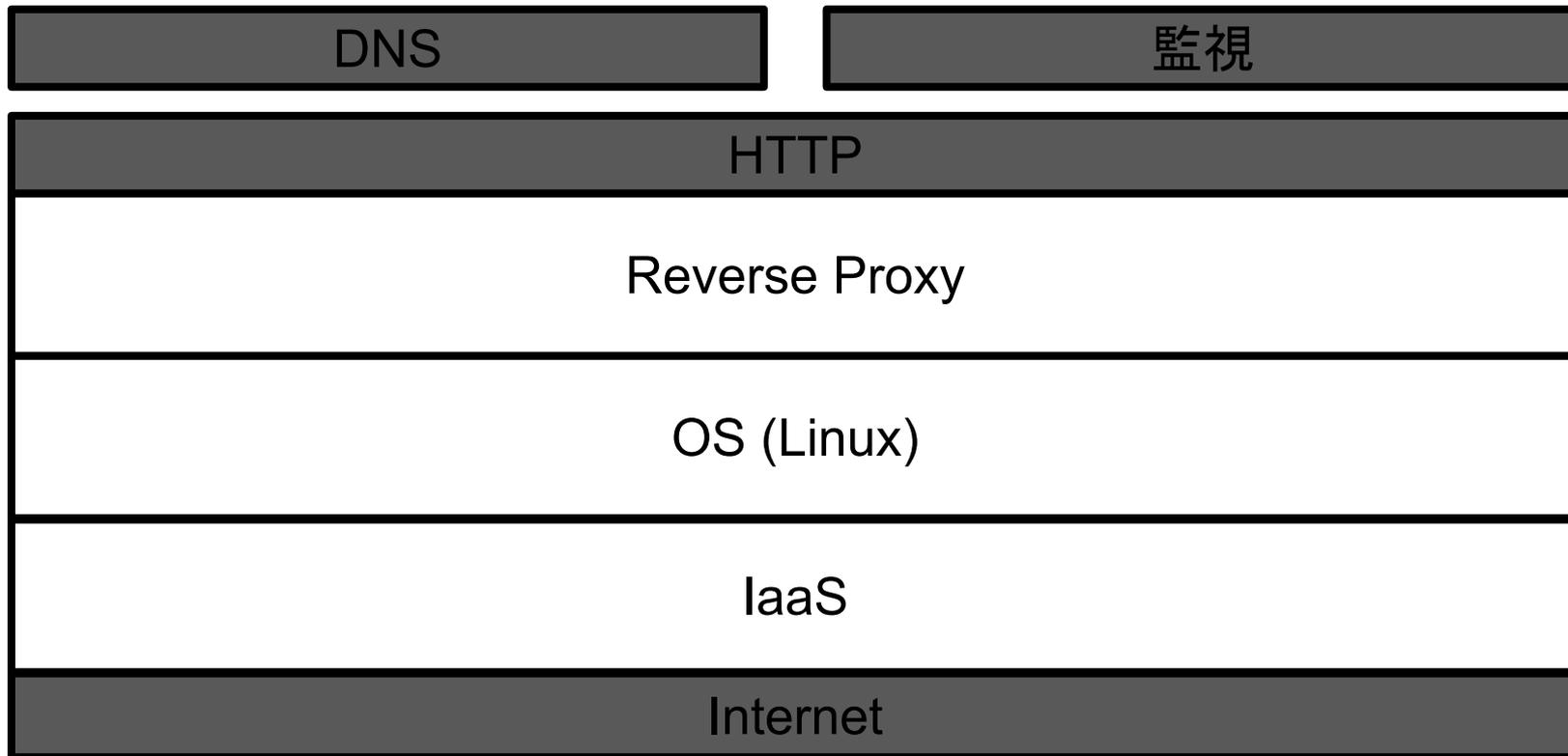
# 自社 Web サービス IPv6 対応の要件



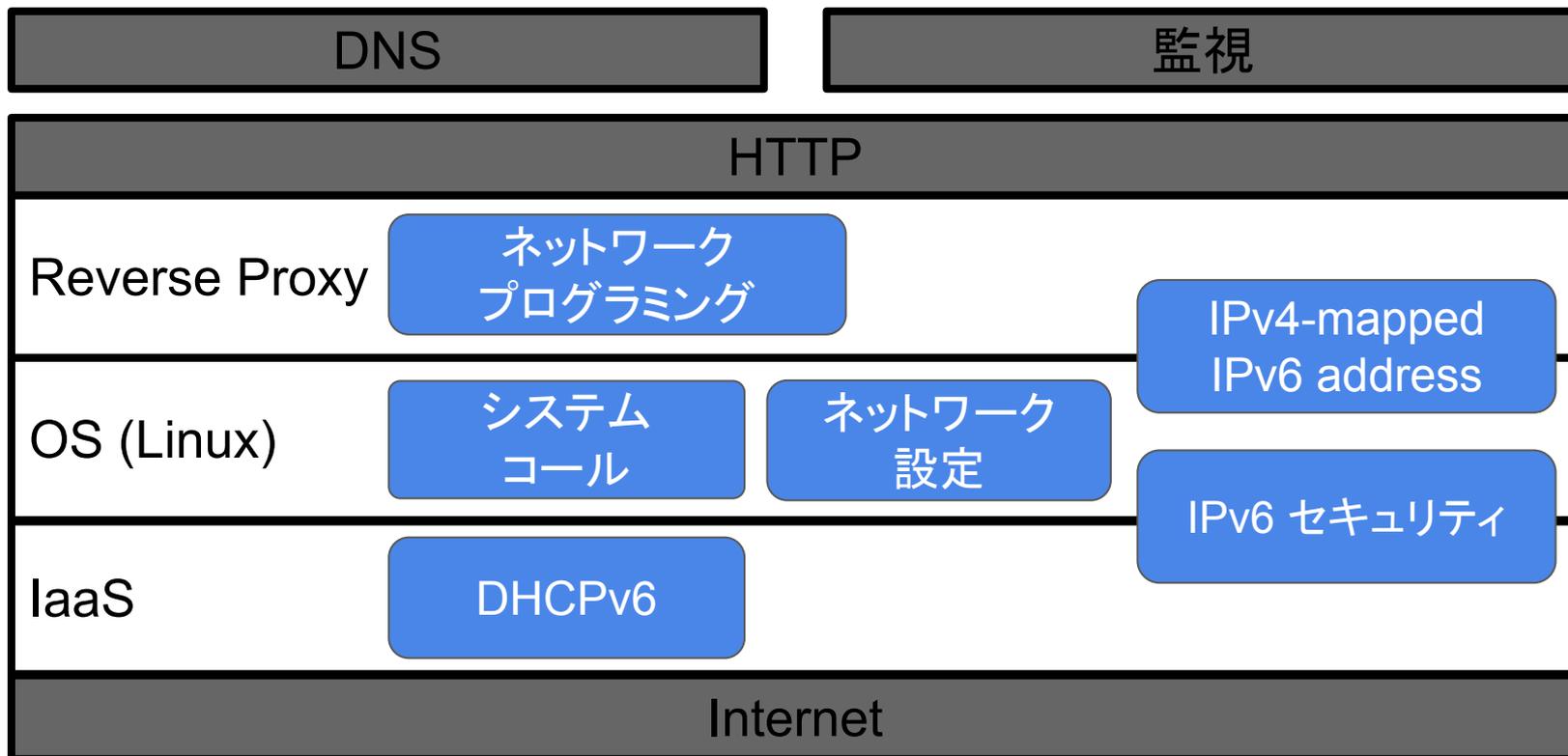
# IPv6 対応のために調査・検証したレイヤ



# 今日お話しするレイヤ



# 今日お話しするレイヤ 詳細



# IPv6 調査・検証の流れ

- IPv6 の基礎を学習する (書籍、RFCなど)
- とにかく IPv6 を使ってみる (Docker で環境構築)
- IPv6 プログラミングしてみる
- IaaS 上の Linux VM で IPv6 を使ってみる

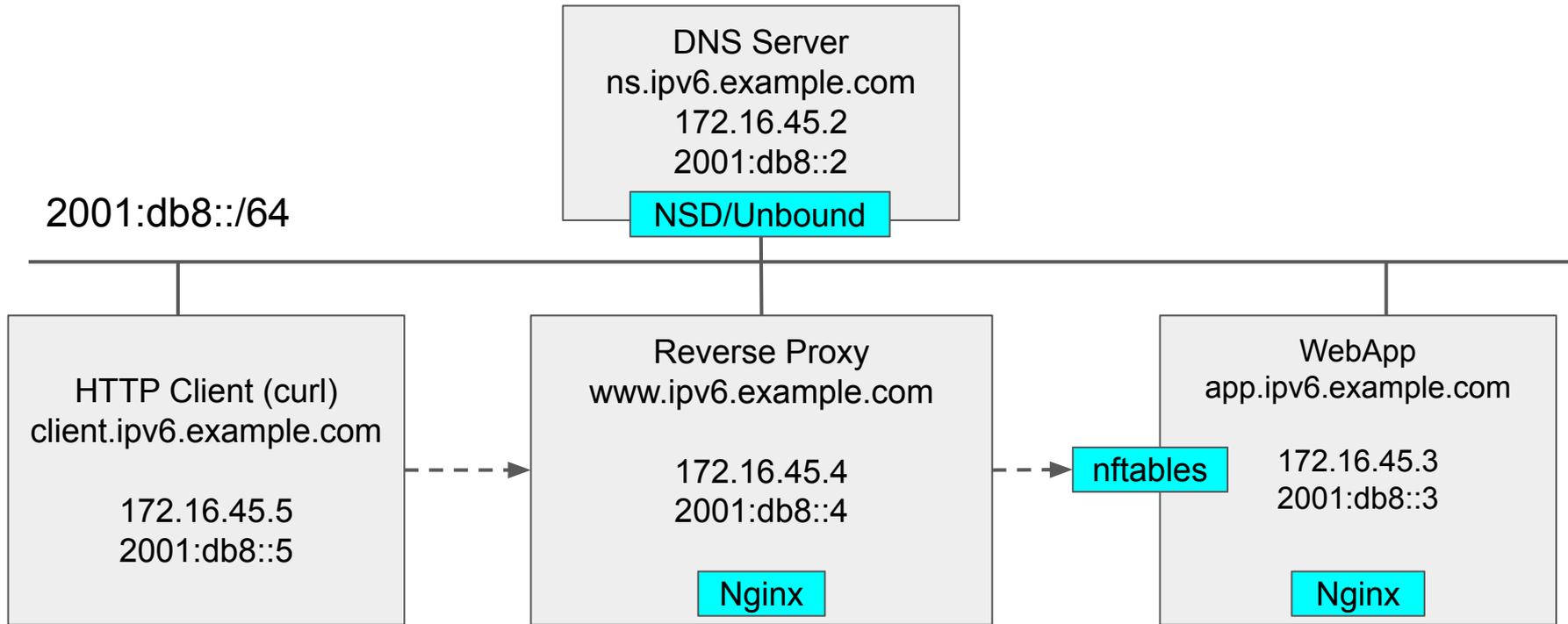
# IPv6 の基礎を学習する

- 書籍を読み込む
  - 小川晃通, プロフェッショナルIPv6 第2版
- RFC、POSIX の仕様を読む
- ここまでの感想:「IPv6 は IPv4 とは完全に違うなあ、覚えることが多いなあ」

# とにかく IPv6 を使ってみる

- 手を動かして試せる環境が欲しい
- => Docker で環境を構築
- オープンソースソフトウェアなどを使って手を動かして IPv6 の世界を体験
- ここまでの感想: 「IPv6 楽しい！」

# Docker で作った IPv6 実験環境



2001:db8::/64 は例示用の IPv6 アドレス。

実際には ULA(Unique Local IPv6 Unicast Addresses) を割り当てた。

# IPv6 プログラミングしてみる

- C 言語で IPv6 ソケットを作ってみる
  - 本来の要件に C 言語プログラミングはないけど、自分でやってみれば理解が深まるだろうと思いやってみる
  - Linux オンラインマニュアル (man)などを参考にした
  - 後々、調査やトラブルシューティングでこの経験が非常に役立った
- Java でプログラミングしてみる
  - ソケットを作ってみる

# IPv6 プログラミングしてみる

- よく参照した Linux オンラインドキュメント
- `getaddrinfo(3)`
- `socket(2)`, `bind(2)`, `listen(2)`, `accept(2)`, `setsockopt(2)`,  
`getsockopt(2)`
- `ip(7)`, `ipv6(7)`, `socket(7)`

# IPv6 プログラミングしてみる

## IPv4-mapped IPv6 address の調査・検証に時間がかかった

- 自前でソケットを作るため、IPv4-mapped IPv6 address を有効にするのか無効にするのか考える必要があった
- 詳細は後述
- ここまでの感想: 「IPv6 は難しいなー」

# laaS 上の Linux VM で IPv6 を使ってみる

- IPv6 対応を検討するサービスは複数の laaS を使ってマルチクラウドでサービスを提供している。調査対象も複数の laaS となる。
- 今回検証した laaS では、IPv6 を有効化することは比較的簡単
  - 最初から有効 (Linux VM 起動直後から IPv6 利用可能)
  - 明示的に有効化
- DHCPv6 などは laaS ごとに違いがあって興味深い
  - IPv4 では気にしたことがないのである意味新鮮

# IaaS 上の Linux VM で IPv6 を使ってみる

- DHCPv6 の IaaS による違い
  - DHCPv6 + RA
  - DHCPv6 (RA 未使用)
  - 同じ IaaS でも、Linux ディストリビューションによっては DHCPv6 での自動割り当てができないものもあったりした
- セキュリティ対策は必要なのかどうか悩んだ(後述)
- ひとまずここまでで技術調査は完了とした
- ここまでの感想: 「IPv6 は沼」

# IPv6 対応検討の結論

- 技術面
  - 技術的には IPv6 には対応可能という結論
  - 既存資産で修正が必要な部分なども把握はできた
  - 時間はそれなりにかかりそう(数ヶ月ぐらい?)
- ビジネス面
  - IPv6 需要はそこまでない。費用対効果なども考えると、いまのところは保留で...
- 調査期間・工数など
  - 半年間ほど
  - 自分でまとめた資料は A4 100 ページほどの量になった

ハマったこと・困ったこと  
よかったこと

# ハマったこと - IPv4-mapped IPv6 address

- IPv4-mapped IPv6 address
  - 簡単にいうと IPv6 ソケットで IPv4 の通信もできる

# IPv4-mapped IPv6 address のソケットの例

IPv4、IPv6 が有効な Linux サーバーで Apache HTTP Server、OpenSSH が起動している

```
$ ss -lnt (出力の一部のみ抜粋)
```

```
0.0.0.0:22
```

```
:::22
```

```
*:80
```

```
$ ss -lnt --ipv4 (IPv4 ソケットのみ)
```

```
0.0.0.0:22
```

```
$ ss -lnt --ipv6 (IPv6 ソケットのみ)
```

```
:::22
```

```
*:80 (IPv4-mapped IPv6 address)
```

# IPv4-mapped IPv6 address のソケットの例

```
# lsof -nP -iTCP -sTCP:LISTEN (出力の一部のみ抜粋)
```

```
COMMAND  TYPE  NAME
```

```
sshd      IPv4   *:22 (LISTEN)
```

```
sshd      IPv6   *:22 (LISTEN)
```

```
httpd     IPv6   *:80 (LISTEN) (IPv4-mapped IPv6 address)
```

```
httpd     IPv6   *:80 (LISTEN)
```

# ハマったこと - IPv4-mapped IPv6 address

- Linux ではソケットオプションの **IPV6\_V6ONLY** で IPv6 ソケットの設定を変更可能
  - IPV6\_V6ONLY オプションが有効: IPv6 のみ通信可能
  - IPV6\_V6ONLY オプションが無効: IPv6 と IPv4 で通信可能
- **注意しなければいけないことがいろいろある**

```
$ ss -ltn -e (出力の一部のみ抜粋)
0.0.0.0:22
[::]:22      v6only:1
*:80        v6only:0
```

# ハマったこと - IPv4-mapped IPv6 address

- IPV6\_V6ONLY の値とソケット作成時に指定する IP アドレスによって挙動が異なる(下表は C 言語の場合)

Address family	IP Address	sockopt	通信可否
AF_INET	Wildcard (0.0.0.0)	-	IPv4
	指定	-	IPv4
AF_INET6	Wildcard (:::)	IPV6_V6ONLY=false (Default)	IPv6 <b>IPv4 (IPv4-Mapped IPv6 address)</b>
	Wildcard (:::)	IPV6_V6ONLY=true	IPv6
	指定	IPV6_V6ONLY=false (Default)	IPv6
	指定	IPV6_V6ONLY=true	IPv6

## ハマったこと - IPv4-mapped IPv6 address

- (ワイルドカードではない) IP アドレスを指定してソケットを作る可能性もあったので、どの組み合わせが自分たちにとって最適か調べる必要があった
- 他の言語だと(例えばJava)、その言語特有の仕様があって、もう少し話が複雑になることがある(詳細は割愛します)

## 困ったこと - IPv6 セキュリティ対策

- 調べていくうちに、不正 RA 対策、ICMPv6 フィルタリングなど  
は何か対策する必要があるのかなと考えた
  - 自分が見た情報では、これらに言及しているものが多かった
- 不正 RA 対策
  - そもそも対策が必要なのかどうか悩んだ
- ICMPv6 フィルタリング
  - RFC 4890 で推奨フィルタリングルールが紹介されているけど、自分たちの環境に適用すべきルールを判断できず

# ハマったこと - 不正 RA 対策

- 不正 RA 対策は必要なのかどうか悩んだ
  - この発表にあたって、IPv4 でも同じようなリスクはあると教えていただいたが、ネットワークは本業ではないので、調査しているときは悩んだ
- Linux のカーネルパラメーター `accept_ra`
  - RA を受け取るかどうか指定できる
  - Linux のネットワーク設定ツール(`netplan(5)`, `interfaces(5)`など)でも `accept_ra` に相当するパラメーターがある
  - `sysctl(8)` で設定した値と、ネットワーク設定ツールの設定値のどちらが適用されているのかわからず確認に時間がかかった

## よかったこと

- 世の中の IPv6 の話題が少しはわかるようになった！
- IPv6 Summit in TOKYO 2022 を視聴
  - ラストセッションのパネルディスカッション。ぶっちゃけた話をされていたように思ったけど、内容が理解できて面白かった。

まとめ

# まとめ

- 自社 Web サービスの IPv6 対応を検討した
- IPv6 は覚えることが多かった(自分メモは A4 で 100 ページぐらいになった)
- 初期の技術調査の段階で時間がかかった
  - 今回はネットワークプログラミングする必要がある、これが時間がかかった一因ではある
- IPv6 セキュリティ対策に悩んだ
- IPv6 は楽しい！

ありがとうございました

# 議論・質疑応答

# 私の疑問

- IPv6 検証環境の作り方
  - みなさんは検証環境はどうやって作っていますか？
- セキュリティ対策
  - どのようなことに気をつければいいのでしょうか？
  - 不正 RA 対策, ICMPv6 フィルタリング
  - IPv4 ではこういうことはあまり気にしていなかった
- ビジネス面
  - IPv6 を使いたいという要望はどれぐらいありますか？