

株式会社 ITS MORE

2020年4月設立

ITS more

2020年6月22日 投稿者: SATOXITS

AWSでARM Linux仮想マシン

開発：AWSでは何度か作っては放置し破棄して来ましたが、そろそろ実用のVMを作りましょう。主な用途はDNSサーバとARMの性能評価。

基盤：そうですね。要件はこれだけですか。

- ・ CPUは ARM
- ・ OSはUbuntu できれば 20.04 LTS
- ・ ディスクは最小限
- ・ 固定IPv4アドレス
- ・ 月間料金2,000円以内

開発：そうですね。

基盤：では始めます。EC2ダッシュボードからかな。あれ？作成というメニューが無いですね？

開発：どうもその「**インスタンスを起動**」ってやつじゃないですか？

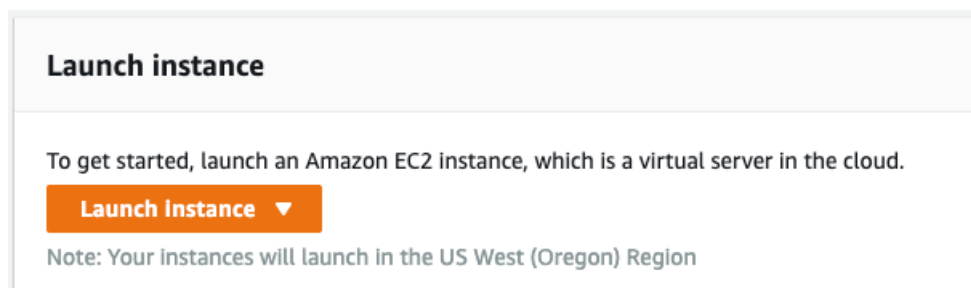
インスタンスを起動

開始するには、クラウド内の仮想サーバーである Amazon EC2 インスタンスを起動します。

インスタンスを起動 ▼

注意: インスタンスは 米国西部 (オレゴン) リージョンで起動されます

基盤：普通起動って言うと、既にあるのを起動するって言う意味ですよ。日本語が変なのかな。英語に切り替えましょう。あ「**Launch instance**」ですね。それなら納得。なんで「作成」とか「作成して起動」って訳さないんでしょう。しかも「開始」じゃなんだかわからないけど「get started」なら感じが出ます。さあ始めましょう的な。まあ進めましょう。

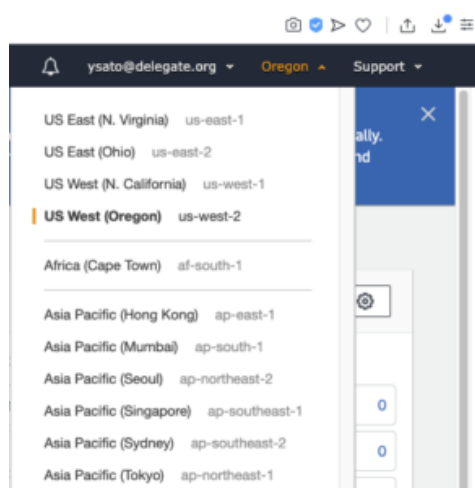


開発：それにしてもなんでオレゴンなんですかね。理由が説明されてない。

基盤：あ、これはアカウントの隣でOregonを選んでもらいたいんですね。これは何を選んでいるのやら、Azureには無いやり方。

開発：メニューなりonmouseoverに、これはなにを選択するものなのか一行説明があればね。Tokyoにしましょう。割高かもしれないですが、Azureもアジアにしてるし、てぷこ支援の一環です。

基盤：電気を大切にね…



Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Launch instance ▼

Note: Your instances will launch in the Asia Pacific (Tokyo) Region

基盤：Launch！てのも変なんですけど。これから Launch の準備するんで。

開発：おっ、Ubuntu 20.04 LTS / ARM ありますね。リストのえらく下に。

Free tier eligible **Ubuntu Server 20.04 LTS (HVM), SSD Volume Type**, EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Architecture: 64-bit (x86) 64-bit (Arm)

基盤：じゃこれを Select ポチッと。

ステップ 2: インスタンスタイプの選択

Amazon EC2 では、異なるユースケースに合わせて最適化されたさまざまなインスタンスタイプが用意されています。インスタンスとは、アプリケーションを実行できる仮想サーバーです。インスタンスタイプはさまざまな CPU、メモリ、ストレージ、ネットワークキャパシティの組み合わせによって構成されているため、使用するアプリケーションに合わせて適切なリソースの組み合わせを柔軟に選択できます。インスタンスタイプおよびそれをコンピューティングのニーズに適用する方法に関する [詳細はこちら](#)。

フィルター条件:

現在選択中: a1.medium (- ECU, 1 vCPU, 2.3 GHz, AWS Graviton Processor, 2 GiB メモリ, EBS のみ)

	ファミリー	タイプ	vCPU	メモリ (GiB)	インスタンスストレージ (GB)	EBS 最適化	ネットワークパフォーマンス	IPv6 サポート
<input checked="" type="checkbox"/>	汎用	a1.medium	1	2	EBS のみ	はい	最大 10 ギガビット	はい
<input type="checkbox"/>	汎用	a1.large	2	4	EBS のみ	はい	最大 10 ギガビット	はい

基盤：さて、どれでしょう。最安ので、2.3GHz、メモリ2GBです。ラズパイクラスですね。でもラズパイと違ってSSD付きだし大丈夫かな？

開発：Azureでは1GBのx86で普通にUbuntu18動いてますしね。

基盤：ではこれで確認と作成。

ステップ 7: インスタンス作成の確認

インスタンスの作成に関する詳細を確認してください。各セクションの変更に戻ることができます。[作成] をクリックして、インスタンスにキーペアを割り当て、作成処理を完了します。

⚠ お客様のインスタンス設定は無料利用枠の対象ではありません

無料利用枠の対象であるインスタンスを起動するには、選択している AMI、インスタンスタイプ、設定オプション、ストレージデバイスをチェックします。無料利用枠の利用枠と使用制限に関する詳細情報をご覧ください。

[今後はこれを表示しない](#)

開発：無料利用枠とかどうでもいいですね。計算がわからなくなる。今後はこれを表示しない、と。

ステップ 7: インスタンス作成の確認

インスタンスの作成に関する詳細を確認してください。各セクションの変更に戻ることができます。[作成] をクリックして、インスタンスにキーペアを割り当て、作成処理を完了します。

AMI の詳細
AMI の編集

Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-061102f51d47b1c24

無料利用枠の対象
Ubuntu Server 20.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).
ルートデバイスタイプ: ebs 仮想化タイプ: hvm

インスタンスタイプ
インスタンスタイプの編集

インスタンスタイプ	ECU	vCPU	メモリ (GiB)	インスタンス ストレージ (GB)	EBS 最適化利用	ネットワークパフォーマンス
a1.medium	-	1	2	EBS のみ	はい	Up to 10 Gigabit

基盤：インスタンスタイプの編集ってなんですかね。プチッ。… げげっ、振り出しに戻ってしまった！？さっき選んだやつをリストの一番上に出してくれないかなあ。

開発：English / 日本語の切り替えでも振り出しに戻ってしまいますね。何なんでしょうこれ。

基盤：8GBのディスクがデフォで付いてますね。これ削れるのかな？

ステップ 4: ストレージの追加

インスタンスは次のストレージデバイス設定を使用して作成されます。インスタンスに追加の EBS ボリュームやインスタンスストアボリュームをアタッチするか、ルートボリュームの設定を編集することができます。また、インスタンスを作成してから追加の EBS ボリュームをアタッチすることもできますが、インスタンスストアボリュームはアタッチできません。Amazon EC2 のストレージオプションに関する [詳細](#) はこちらをご覧ください。

ボリュームタイプ	デバイス	スナップショット	サイズ (GiB)	ボリュームタイプ	IOPS	スループット (MB/秒)	終了時に削除	暗号化
	/dev/sda1	snap-0da7df2dd1e9a3691	4	汎用 SSD (gp2)	100 / 3000	該当なし	<input checked="" type="checkbox"/>	暗号化

⚠ ボリュームサイズは、少なくともスナップショットのサイズ (8 GiB) にする必要があります。

基盤：だめでした。なら選ばせるなって一の。SSDじゃなくてHDDに出来ますかね？



開発：何言ってるんだかわかりません。HDDを選んだら使いものになるのかとか、おいくらくらい違うのか、知りたいのはそこなんですけど。

基盤：Azure で仮想マシンを作る時は、今の構成ですとおいくらですっ、っていつも言ってくれるんですけどね。

開発：ところがどっこい、オプションで付けたディスクの値段を加算してくれないので勘違いしてひどい目にあう人も出る（笑）

基盤：そもそもこないだ試作した t2.micro ってやつだと、立ち上がりが40秒とか50秒とかかかりました。あれは話にならないですね。Azureのb1sだって30~40秒で起動するんですが。

開発：まあ、デフォのgp2というのにしときましよう。8GBだし、そうばられないでしょう。

- 1. AMI の選択
- 2. インスタンスタイプの選択
- 3. インスタンスの設定
- 4. ストレージの追加
- 5. タグの追加
- 6. セキュリティグループの設定
- 7. 確認

ステップ 3: インスタンスの詳細の設定

要件に合わせてインスタンスを設定します。同じ AMI からの複数インスタンス作成や、より低料金を実現するためのスポットインスタンスのリクエスト、セス管理ロール割り当てなどを行うことができます。

インスタンス数 [Auto Scaling グループに作成する](#)

購入のオプション スポットインスタンスのリクエスト

ネットワーク [新しい VPC の作成](#)

サブネット [新しいサブネットの作成](#)

自動割り当てパブリック IP

配置グループ インスタンスをプレイズメントグループに追加します。

キャパシティの予約 [新しいキャパシティ予約の作成](#)

IAM ロール [新しい IAM ロールの作成](#)

シャットダウン動作

終了保護の有効化 誤った終了を防止します

モニタリング CloudWatch 詳細モニタリングを有効化
追加料金が適用されます。

EBS 最適化インスタンス EBS 最適化インスタンスを作成する

テナンシー
Additional charges may apply when launching Dedicated instances.

Elastic Inference Elastic Inference アクセラレーターを追加
追加料金が適用されます。

ファイルシステム [新しいファイルシステムの作成](#)

▼ 高度な詳細

Metadata accessible

Metadata version

Metadata token response hop limit

ユーザーデータ テキストで ファイルとして 入力はすでに base64 でエンコード済み
(省略可能)

開発：基本、全てデフォルトで良さそうです。というか選択の余地が無い。

- 1. AMI の選択
- 2. インスタンスタイプの選択
- 3. インスタンスの設定
- 4. ストレージの追加
- 5. タグの追加
- 6. セキュリティグループの設定
- 7. 確認

ステップ 6: セキュリティグループの設定

セキュリティグループは、インスタンスのトラフィックを制御するファイアウォールのルールセットです。このページで、特定のトラフィックに対してインスタンスへの到達を許可するルールを追加できます。たとえば、ウェブサーバーをセットアップして、インターネットトラフィックにインスタンスへの到達を許可する場合、HTTP および HTTPS ポートに無制限のアクセス権限を与えます。新しいセキュリティグループを作成するか、次の既存のセキュリティグループから選択することができます。Amazon EC2 セキュリティグループに関する [詳細はこちら](#)。

セキュリティグループの割り当て: 新しいセキュリティグループを作成する
 既存のセキュリティグループを選択する

セキュリティグループ名:

説明:

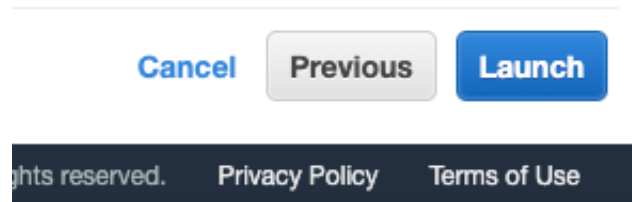
タイプ	プロトコル	ポート範囲	ソース	説明
SSH	TCP	22	カスタム 0.0.0.0/0	例: SSH for Admin Desktop

[ルールの追加](#)

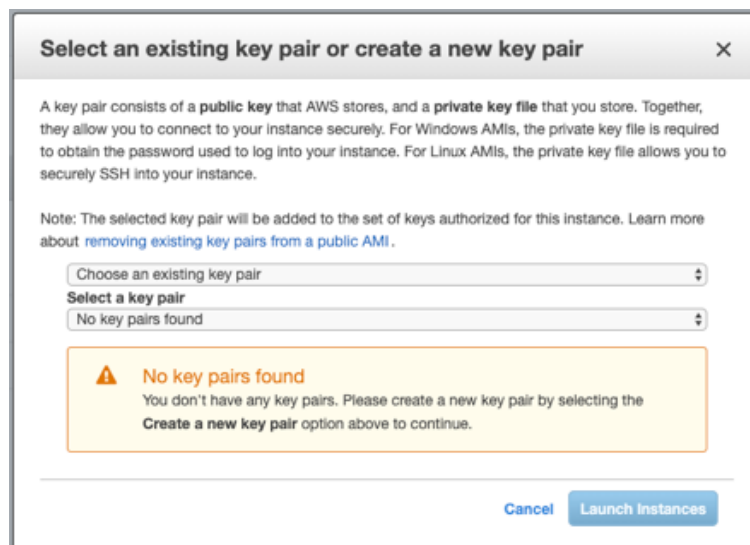
警告
送信元が 0.0.0.0/0 のルールを指定すると、すべての IP アドレスからインスタンスにアクセスすることが許可されます。セキュリティグループのルールを設定して、既知の IP アドレスからのみアクセスできるようにすることをお勧めします。

開発：それでは「起動」ボタンを…

基盤：なんか違和感あるなあ。英語に切り替えてやり直し。Launch をポチッ！



基盤：



開発：ここでこれを聞いてきますか。

基盤：若干シラケますね。じゃま、心機一転、鍵ペアを生成して、Launch Instance
！

ap-northeast-1.console.aws.amazon.com/ec2/v2/home

aws Services Resource Groups

Launch Status

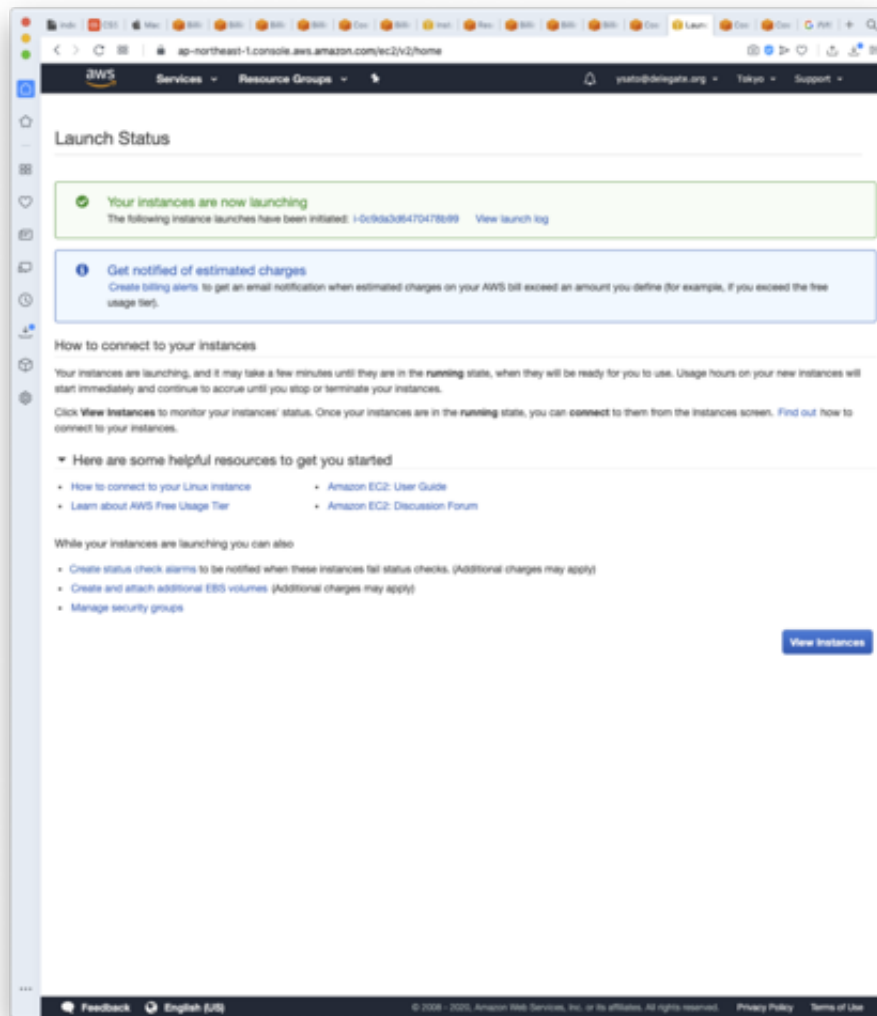
✔ **Your instances are now launching**
The following instance launches have been initiated: [i-0c9da3d6470478b99](#) [View launch log](#)

ℹ **Get notified of estimated charges**
Create [billing alerts](#) to get an email notification when estimated charges on your AWS bill exceed a usage tier).

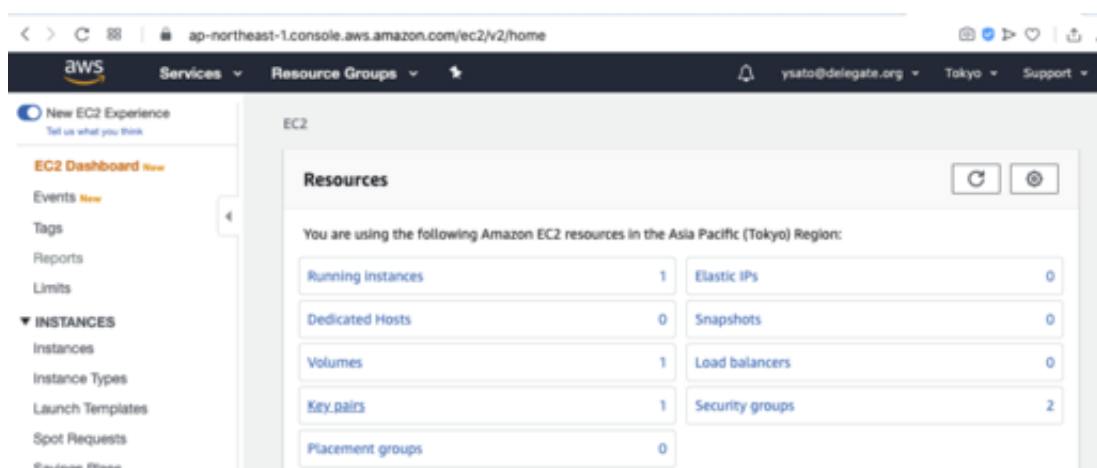
How to connect to your instances

開発：これって確か5分くらいかかりまよね。

基盤：この画面で忘れちゃうんですよね。キャプチャ。



基盤：あ、この画面消してしまいました。再表示する方法がわからん…



開発：Azure だとバックグラウンドで何やってるかいつも表示されてますよね。

基盤：プロビジョニングがどうか。あれはあれでうざいんですけどね。カタカナに変換するだけなら機械でもできるだろって。

開発：Google翻訳によると provisioning は中国語で「供應」ですね。で人語「供应」をGoogleさんは「Servant」と訳す。お・も・て・な・し？こういうの、一つの単語で表現しようとするのが無理なんですよね。

基盤：ここに出てくる running が1つって、以前作ったやつのことですよ。

開発：なんかもう15分くらい経ったけど出来てこないですね… あ、メールが。



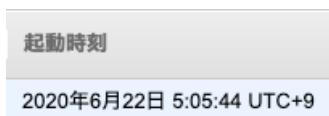
基盤：何言ってるんだか、さっぱりわかりませんね。「お客様は」って私らのことですか？そんなリクエストした覚えはありませんが。

開発：「お客様のリクエストが検証されました」って、検証に成功したって意味ですかね？

基盤：「引き続き問題が発生する場合」ってありますから、何か問題が発生したんでしょうね。いつどういう操作をして何が起きたみたいなログは見られないんですかね。

開発：あ。これって、Tokyo リージョンには running が1つって意味じゃないですか？

基盤：確かに。オレゴンに一つ。東京に一つ。東京で実行中のインスタンスは…



基盤：おお、さっき立ち上がったってわけですね。じゃあこれか。

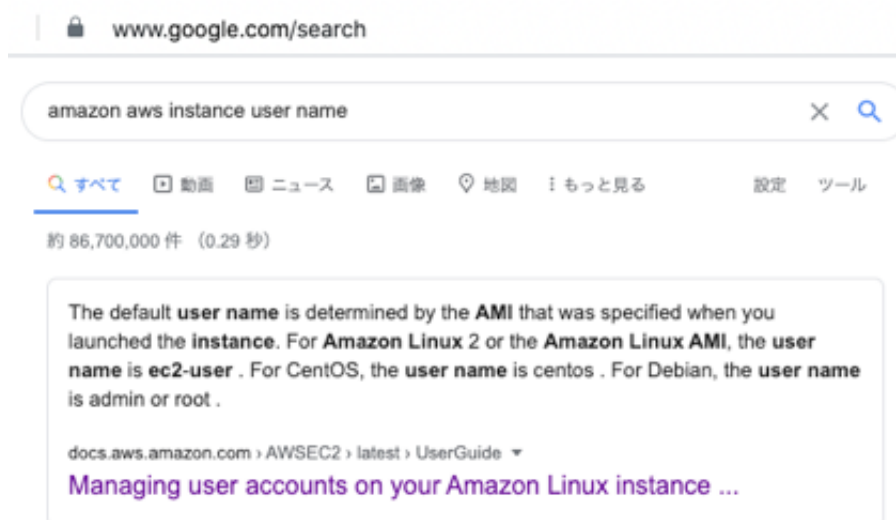
開発：SSHでログインしましょう。

基盤：あれ？ログイン名って何なんでしょう？ Azureだと自分で決めるのですが。

開発：キーペアを提供しているところになんか説明がないですかね。

基盤：無いですね。AWSのヘルプを検索してもよくわかりませんね… なんだあこの検索は。具体的には何なんですかねこの instance-user-name で文字列は。

開発：ぐぐりましょう。aws instance user name …



開発：答え一発カシオミニ。ですが ubuntu とか一言も。meta description のせいですかね。

基盤：そのページの実体、こうなってます。

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/managing-users.html>
Managing user accounts on your Amazon Linux instance

Each Linux instance launches with a default Linux system user account. The default user name is determined by the AMI that was specified when you launched the instance. For Amazon Linux 2 or the Amazon Linux AMI, the user name is `ec2-user`. For CentOS, the user name is `centos`. For Debian, the user name is `admin` or `root`. For Fedora, the user name is `ec2-user` or `fedora`. For RHEL, the user name is `ec2-user` or `root`. For SUSE, the user name is `ec2-user` or `root`. For Ubuntu, the user name is `ubuntu`. Otherwise, if `ec2-user` and `root` don't work, check with your AMI provider.

基盤：EC2、ドキュメントが腐ってますね。儲かってないのかな？

開発：あなたのところでは Fedora だ CentOS だって今もサポートしているんすかって20ぺんくらい聞いてみたいですね。まあでも、ユーザ名は `ubuntu` だ、ということ

基盤：やれやれ。

```
MacMini% ssh -i aws-xxx-xxx.pem ubuntu@18.180.xxx.xxx
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-1009-aws aarch64)
...

$ uptime
20:47:25 up 41 min,  1 user,  load average: 0.11, 0.03, 0.01
```

基盤：起動してからログインするまで41分も無駄にしてみました。まあでも、普通っぽいですね。ディスクも十分空いていると思います。

開発：Azure の Ubuntu っていきなり8GB以上詰まってませんでしたっけ？

```
$ df /
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/root          7959716 1295456   6647876  17% /
```

基盤：とりあえず `g++` だけはインストールしましょう。 `sudo apt install make g++`
...

```
$ df /
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/root          7959716 1423948   6519384  18% /
```

基盤：無問題ですね。でとりあえず `cc hello.c`。

```
$ ./a.out  
Hello World!
```

開発：OKですね。

基盤：いやはやしかし、時間を無駄にしてみました。さてそれではめっちゃ楽しみにしていた DeleGate の make 所要時間測定を。。むむ一、これは遅いですね。ラズパイと同等なの？…

```
real    4m32.267s  
user    4m4.947s  
sys     0m25.348s
```

開発：うーむ。ラズパイ4で6分でしたから、30%以上は速いということですかね。ほぼCPUタイム。では、openssl で乱数の生成ベンチマーク。

```
arm@aws$ time openssl rand 1000000000 > /dev/null
```

```
real    0m3.307s  
user    0m3.066s  
sys     0m0.240s
```

基盤：一方その頃 Azure 上の x86 では。

```
x86@azure$ time openssl rand 1000000000 > /dev/null
```

```
real    0m3.596s  
user    0m3.321s  
sys     0m0.088s
```

開発：なるほど。コンパイラが使うようなCPU命令はx86に比べて遅いけど、randが使う命令はx86と同等ってことかな。あるいはキャッシュのサイズとアプリのメモリアクセスの相性とか。まあうちの的にはこの性能で良さそうですね。少なくとも DNS サーバとして使うには無問題。ところでリブートは？

基盤：50秒以上かかりますね。1分半かかったり。どうも AWS は、少なくとも安物マシン同士で比べると、起動が Azure に比べて2倍近く遅い感じがあります。再起動が60秒以上というのは、使い方に制約がかかりますね。これはあきらめるしかないような。

基盤：無駄なところで疲れてしまいました。料金はあとで確認しましょう。いずれにし

でもこれが実質最少料金。デフォの8GB SSDにもVM料金とは別に課金があるのかということと、固定 IP アドレスの料金ですね。

開発：でもまあ一日50円では行けそうな。

— 2020-0622 SatoxITS

[AWSでARM-Linux仮想マシン—株式会社-ITS-more-2](#)

ダウンロード