



BE A SOLID.

検証から分かった
NGINX Plusの魅力とは？

自己紹介

角田 竜馬

株式会社grasys
Cloud infrastructure Division
Ops Team

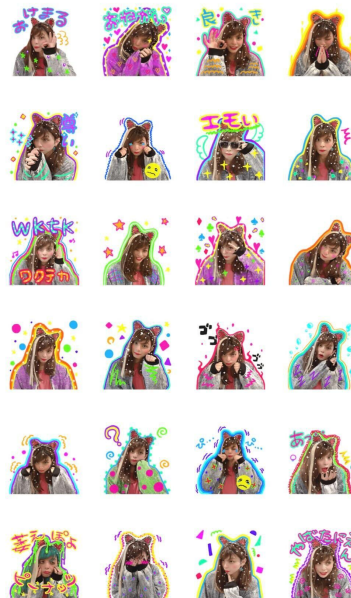
2018年10月に grasys へ入社。現在はプレイヤーとしてインフラ構築・運用を行っている。

grasys 入社以前は潜水士をしたりイラストレーター兼特殊塗装業をしていた。

自己紹介

LINEスタンプ「かなめちゃんのスタンプ☆」ございます。

かなめちゃんのスタンプだお。



TATSUMA

本当に何もやることが無い時はかなめちゃんに変身して遊びます。

会社紹介



「もっと強固なインフラに」

社名	株式会社 grasys
創業日	2014 / 11 / 13
代表	長谷川 祐介
資本金	1,000 万円
社員数	40 名
所在地	恵比寿

会社紹介:事業規模

エンドユーザー数	累計 3 億超ユーザー
クラウドプロジェクト数	200 プロジェクト
VM インスタンス運用実績	4,500 台／月
最大稼働インスタンス数	2,200 インスタンス／システム
1 秒間のリクエスト回	200 万回／秒
ビッグデータの分析基盤	120 兆レコード／日
データストリーミング分析	2,000 ノード
分散データベース	280 ノード

会社紹介: パートナー



クラウドインフラのオーケストレーションプラットフォーム



インスタントページが特徴のエッジクラウドプラットフォーム



検索・監視・セキュリティのオールインワンプラットフォーム



脆弱性対策ソリューションを備えたプラットフォーム



クラウドネイティブセキュリティプラットフォームを擁するセキュリティ業界の巨人



ビジネスインテリジェンス(BI)データプラットフォーム

NGINX Plus検証経緯

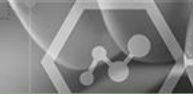




検証から分かったNGINX Plusの魅力とは？

ポイント

- 機能や設定の方法は軽く、こう使ったら良いかもね。
- gcpのhealthcheckではなく、active health checkの何が良いのか。
- こういう場合はNGINX Plusが良いよね。
- 検証結果を元に良かった部分
- 何と比べて魅力なのか
- 例えば設定が簡単っていうのは、なんで簡単なのか。



検証環境

検証期間: 2022/ ~ 2022/

Google Cloud

サーバー: Compute Engine

OS: Cent OS 7

Region: asia-northeast1(東京)



NGINX+とは

NGINX Plusは、オープンソース版のNGINXに様々な機能を拡張し搭載した商用版ソフトウェア。

ロードバランサー、セキュリティ、冗長化構成など、各機能に魅力的な拡張機能が搭載されている。

サポートもついており、導入支援や技術QAに加え、脆弱性に対応するパッチ提供もあり安心して使用できる。

詳しくは東京エレクトロデバイス様のブログで紹介しているのでご参照ください。

※料金については東京エレクトロデバイス様にお問い合わせお願いします笑



NGINX Plusインストール方法

まずは、https://cs.nginx.com/repo_setup へアクセス

↓

インストールするOSを選択

↓

すると...

NGINX Plusインストール方法

NGINX

PRODUCTS

SOLUTIONS

RESOURCES

SUPPORT

PRICING

BLOG

Q

FREE TRIAL

CONTACT US

To show setup instructions please choose your OS and distribution:

RHEL 7.4+/CentOS 7.4+/Oracle Linux 7.4+ ▼

Installation instructions for RHEL 7.4+ / CentOS 7.4+ / Oracle Linux 7.4+

- If you already have old NGINX packages in your system, back up your configs and logs:

```
sudo cp -a /etc/nginx /etc/nginx-plus-backup
sudo cp -a /var/log/nginx /var/log/nginx-plus-backup
```

- Create the /etc/ssl/nginx/ directory:

```
sudo mkdir -p /etc/ssl/nginx
```

- Log in to [MyF5](#), if you purchased subscription, or follow the link in the trial activation email, and download the following two files:

```
nginx-repo.key
nginx-repo.crt
```

- Copy the above two files to the RHEL/CentOS/Oracle Linux server into /etc/ssl/nginx/ directory. Use your SCP client or other secure file transfer tools.

- Install prerequisite packages:

```
sudo yum install ca-certificates
```

- Install app-protect prerequisite packages:

```
sudo yum install epel-release
```

- Add NGINX Plus repository by downloading the file [nginx-plus-7.4.repo](#) to /etc/yum.repos.d:

```
sudo wget -P /etc/yum.repos.d https://cs.nginx.com/static/files/nginx-plus-7.4.repo
```

- If you have app-protect subscription, add app-protect repositories by downloading the file [app-protect-7.repo](#) to /etc/yum.repos.d:

```
sudo wget -P /etc/yum.repos.d https://cs.nginx.com/static/files/app-protect-7.repo
```

- If you have modsecurity subscription, add modsecurity repository by downloading the file [modsecurity-7.repo](#) to /etc/yum.repos.d:

```
sudo wget -P /etc/yum.repos.d https://cs.nginx.com/static/files/modsecurity-7.repo
```

- Install the NGINX Plus package

```
sudo yum install nginx-plus
```

手順が詳細に記載されており、
分かりやすい。

grasysが魅力を感じた機能はこれ！

- WAF(NGINX App Protect)
- Active Health Check
- NGINX Instance Manager & NGINX Controller
- Ingress Controller
- Service Mesh (もしくはDNSディスカバリー)
- おまけ

WAF (NGINX App Protect)

NGINX Plusを利用しているとAdd-onという形でNGINX App Protectの機能を有効化できる。

設定例 ↓

```
server {  
    listen      80;  
    server_name localhost;  
  
    access_log /var/log/nginx/access.log main;  
  
    location / {  
        app_protect_enable on;  
        root   /usr/share/nginx/html;  
        index  index.html index.htm;  
    }  
}
```

WAF (NGINX App Protect)

```
server {  
    listen      80;  
    server_name localhost;  
  
    access_log /var/log/nginx/access.log main;  
  
    location / {  
        app_protect_enable on;  
        app_protect_policy_file "/etc/nginx/AppProtectTestPolicy.json";  
        app_protect_security_log "/etc/nginx/log-default.json" syslog:server=127.0.0.1:515;  
  
        root /usr/share/nginx/html;  
        index index.html index.htm;  
    }  
}
```


WAF (NGINX App Protect)

反映が早い

	banされるまでの時間	unbanされるまでの時間
GCP Cloudarmor	約2～3分	約2～3分
AWS WAF	約30秒	約5分
NGINX App Protect	即時	即時

例えばDOS攻撃の場合、反映までにサーバーに負荷がかかる。

NGINX App Protectであれば即時反映で負荷を最小限に抑えられる。

WAF (NGINX App Protect)

- fail2ban+NGINX Plusの機能kvsのサーバ間共有を使用して、ban対象のIPを検知したらそのIPをkvsへ登録、NGINXの設定でkvsに登録されているIPは拒否という設定にしておけばbotなどによるDOS攻撃を迅速に防げる。kvsではなくリストファイルを作成、そこに登録でも良い。図作成。
- LB設定のnginxサーバにfail2banを導入、banを検知でnginx kvsにIPを登録、kvsに登録されているIPは拒否という設定にしておけば即時反映。

WAF (NGINX App Protect)

他にも・・・

- DOSやIP制御だけでなく、XSS攻撃などにも対応。
- IP登録制限もcloudarmorでは1ルール10IPと制限があるが、NGINXの場合は特に上限はなくサーバスペックによるのも良い。
- `app_protect_policy_file`の中でポリシーの設定もいじれるので、自由が効く。
- クラウドベンダーのwaf機能だと対象がLBなど大きいため、パス毎の詳細な設定ができないが、NGINX PlusのAPP Protect機能であればlocation毎にポリシー設定ができる。

Active Health Check

アップストリームサーバの正常性を色々な条件でチェックできる。

様々な問題を検出して回避することができ、アプリケーションの信頼性を向上させることができる。

Active Health Check

```
server {  
    location / {  
        proxy_pass http://backend;  
        health_check interval=2s  
            fails=2  
            passes=5  
            uri=/  
            match=healthcheck;  
    }  
}  
  
match healthcheck {  
    status 200;  
    header Content-Type = "text/html";  
    body ~ "Welcome to nginx!";  
}
```

- 2秒ごとに uri/ に対してGET要求を実施
- 5回連続でチェックに合格して正常(passes=5)
- 2回連続失敗で不健全(fails=2)

応答内容はmatchブロックと一致している必要がある

- ステータスコードは200
- ヘッダーのContent-typeはtext/html
- 文字列 Welcome to nginx! を含むbodyを持つ応答

Active Health Check

- 例えばGCPのhealthcheckは、レスポンス、ヘッダー、リクエストパスの設定ができます。文字列も指定できるが制限があるため、より自由なヘルスチェックが可能なNGINX Plusは魅力。
- 弊社のプロジェクトでも、「とある文字列まで拾ってOKとしたい」という要望も受けたことがありますので、レスポンスだけではなく、詳細な文字列取得などの場合は使用できる。いらすとや入れる

Active Health Check

- サーバの健全性を常に保証できるし、NGINX Plusのwafと組み合わせればアタックのbanにも迅速に対応できてセキュリティもより強固にできて良い。さっきのwafの図にactive health checkも加える。

NGINX Instance Manager & NGINX Controller

- 取得できるメトリクスが多い
- group毎にデプロイ可能
- 開発担当者主導でconfigをGUI上から設定、セキュリティ的にsshさせたくない場合などには有用だと思う。
- templateとして履歴は残るが、前回の設定との差分がわからないのでわかるようになるといい。できればconfig適用前に差分がわかりたい。
- GUI画像

Ingress Controller



Service Mesh (もしくはDNSディスカバリー)



番外編:nginx unit (ボリュームによる)

- 色々な言語が動く。pythonいけるしjavaもgoもphpもいける。
- そのため、php-fpmとuwsgiの両方を入れなくて済む。
- 無停止で設定変更可能。

まとめ

- 魅力を語る
- クラウドとの組み合わせも語る
- クラウドを使用していると、そのクラウドのロードバランサを使用することが多いが、マルチクラウドやオンプレ環境で有用である機能もあり、うまく組み合わせられる場面はあると思う。
- 上手く組み合わせられる場面を具体的に。こういう風に組み合わせるとすごく良いかもなっていうのを話す。

宣伝告知などあれば

- blogリレーしてることとか
- 本！





ご視聴ありがとうございました