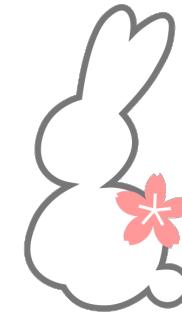




HashiCorp

Terraform



usacloud

ハンズオン for さくらのクラウド ～Usacloud編～

2017/9/26

@yamamoto_febc

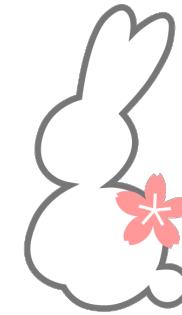
タイムテーブル

開始時刻	概要	内容
19:10～	概要説明(座学)	<ul style="list-style-type: none">Infrastructure as Codeとはusacloudの紹介Terraformの紹介
19:40～	環境構築 (休憩含む)	<ul style="list-style-type: none">クーポン適用さくらのクラウドAPIキー取得/設定
20:00～	usacloud編	<ul style="list-style-type: none">ハンズオン
21:00～	Terraform編	<ul style="list-style-type: none">ハンズオン



HashiCorp

Terraform



usacloud

環境構築編

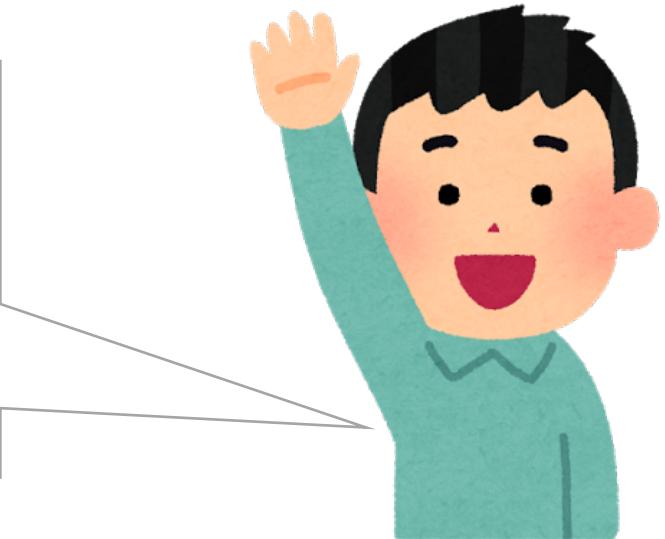


クーポンの適用

クーポン適用マニュアル

<https://manual.sakura.ad.jp/cloud/payment/coupon.html#coupon-apply>

クーポン適用方法がわからない場合
声をかけてください！！





APIキーの取得

usacloud 導入ガイド

https://sacloud.github.io/usacloud/start_guide/#api



「他サービスへのアクセス権」は
請求情報にチェックを！！



APIキーの取得

APIキーを追加

名前* 任意の値を入力

説明 「作成・削除」を選択

アクセスレベル* 無効 リソース閲覧 電源操作 設定集 作成・削除

他サービスへのアクセス権 請求情報 ウェブアクセラレータ

「他サービスへのアクセス権」は
請求情報にチェックを入れる！！



APIキーの取得

webaccel
APIキー (詳細表示)

リソース ID	[REDACTED]
名前	webaccel
説明	webアクセラレーター関連テスト用
ACCESS TOKEN	b[REDACTED]21
ACCESS TOKEN SECRET	NUL[REDACTED]oPQib[REDACTED]i84g
アクセスレベル	リソース閲覧
他サービスへのアクセス権	ウェブアクセラレータ
作成日時	[REDACTED]

アクセストークン
アクセスシークレット



この画面を開いたままにしておいてください
(後で使います)



APIキーの設定



いくつか設定方法がありますが、
今日は 「環境変数」 を使います！！



APIキーの設定

Unix系OSなどの場合

```
# アクセストークン  
$ export SAKURACLOUD_ACCESS_TOKEN=入力
```

```
# アクセスシークレット  
$ export SAKURACLOUD_ACCESS_TOKEN_SECRET=入力
```

```
# デフォルトゾーン(石狩第2を指定)  
$ export SAKURACLOUD_ZONE=is1b
```



APIキーの設定

Windowsの場合

```
# アクセストークン  
$ setx SAKURACLOUD_ACCESS_TOKEN 入力
```

```
# アクセスシークレット  
$ setx SAKURACLOUD_ACCESS_TOKEN_SECRET 入力
```

```
# デフォルトゾーン(石狩第2を指定)  
$ setx SAKURACLOUD_ZONE is1b
```

※設定後、コマンドプロンプトの再起動を行ってください。



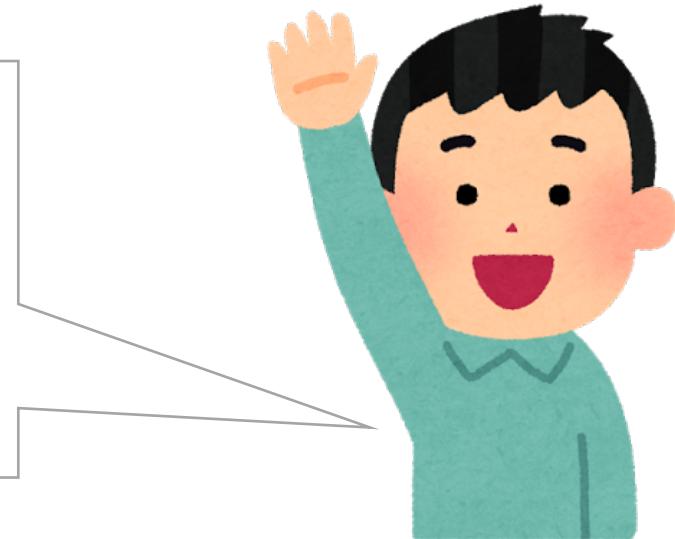


インストール

usacloud 導入ガイド

https://sacloud.github.io/usacloud/start_guide/

インストール方法に不安のある方は
声をかけてください！





インストールのポイント



どれを使って
インストールすればいいの？

利用できる場合はbashと
パッケージ管理システム^(※)がオススメ！

※yum、apt、brew、chocoなど

💡 bashであれば入力補完(bash_completion)を利用可能 💡





インストールの確認

```
# ヘルプを表示  
$ usacloud --help
```



インストールできていれば
ヘルプが表示できるはず



ログイン情報(APIキー)の設定

APIキーは以下3つの方法で指定できます。

1. 環境変数
2. 設定ファイル(configサブコマンド)
3. コマンドラインオプション

ハンズオンでは1.の方法を利用します。
その他の方法についてはドキュメントを
参照してください。



参考：Usacloud 導入ガイド - APIキーの設定

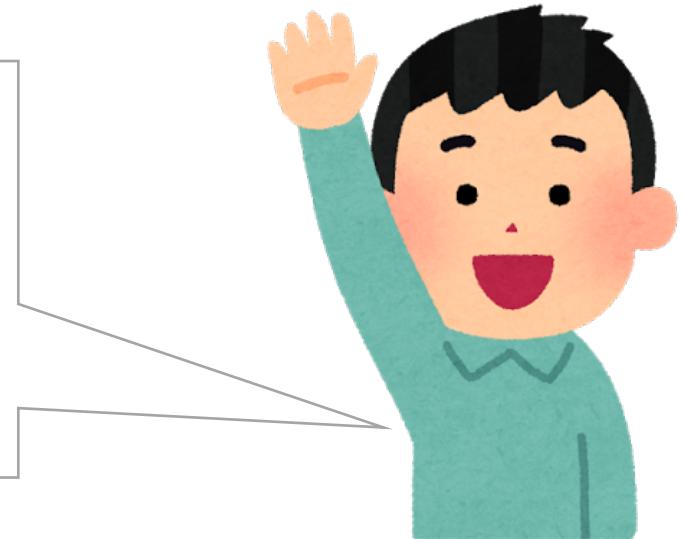
https://sacloud.github.io/usacloud/start_guide/#api_1

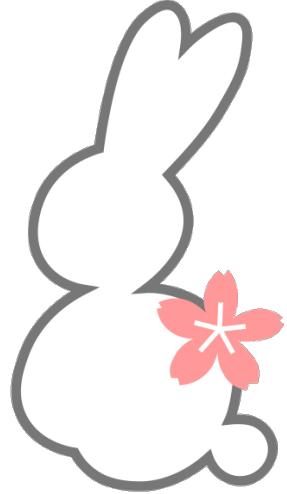


APIキー設定の確認

```
# 認証情報を表示  
$ usacloud auth-status
```

設定が正しく行えている場合、
アカウント情報が表示されるはずです。





usacloud

基本的な使い方



```
$ usacloud [グローバルオプション] <操作対象のリソース> <コマンド> [コマンドオプション] 引数
```

名称	説明
グローバルオプション	APIキーや対象ゾーンなど、全コマンドで共通的に使うオプション
操作対象のリソース	サーバやディスクなど、さくらのクラウド上のリソース
コマンド	一覧(list)、作成(create)、更新(update)などの操作
コマンドオプション	コマンドに対するオプション
引数	操作対象となるリソースのIDや名称など



コマンド構文

この3つが基本

```
$ usacloud [グローバルオプション] <操作対象のリソース> <コマンド> [コマンドオプション] 引数
```

名称	説明
グローバルオプション	APIキーや対象ゾーンなど、全コマンドで共通的に使うオプション
操作対象のリソース	サーバやディスクなど、さくらのクラウド上のリソース
コマンド	一覧(list)、作成(create)、更新(update)などの操作
コマンドオプション	コマンドに対するオプション
引数	操作対象となるリソースのIDや名称など



具体例

```
$ usacloud [グローバルオプション] <操作対象のリソース> <コマンド> [コマンドオプション] 引数
```

```
# サーバの情報を一覧表示(引数なし)  
$ usacloud server ls
```

```
# サーバの情報を一覧表示 + オプション(Nameでソート)  
$ usacloud server ls --sort Name
```



ヘルプの表示方法

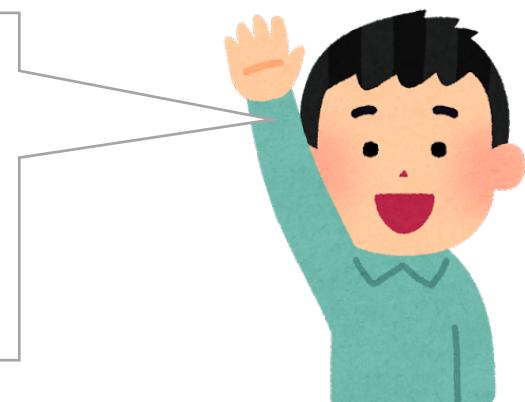
```
# 操作対象リソースについてのヘルプ  
$ usacloud -h
```

```
# 対象リソースのコマンドについてのヘルプ  
$ usacloud server -h
```

```
# コマンドのオプション/引数についてのヘルプ  
$ usacloud server ls -h
```

“--help” または “-h”を指定することで
ヘルプを表示できます。

指定する位置でヘルプ表示も変わるよ！





[TIPS] bashでの入力補完



bashでusacloudをご利用されている場合、TABキーによる入力補完が可能です。

```
$ usacloud [TAB×2]
```

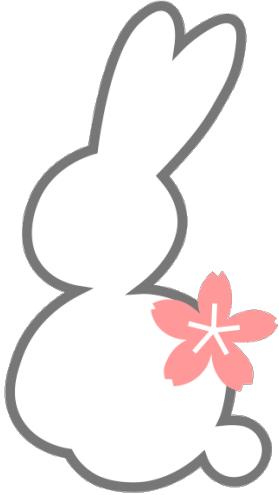
← リソース名を補完

```
$ usacloud --[TAB×2]
```

← オプション名を補完

```
$ usacloud --zone [TAB×2]
```

← オプションの値を補完



usacloud

一覧/CRUD操作



リソースの作成

ハンズオン用にサーバを3台作成します。

- ✓ OSはCentOS
- ✓ 1CPU/1GBメモリ(デフォルト)
- ✓ ディスクはSSH20GB(デフォルト)
- ✓ 名前はそれぞれ「server01」 「server02」 「server03」



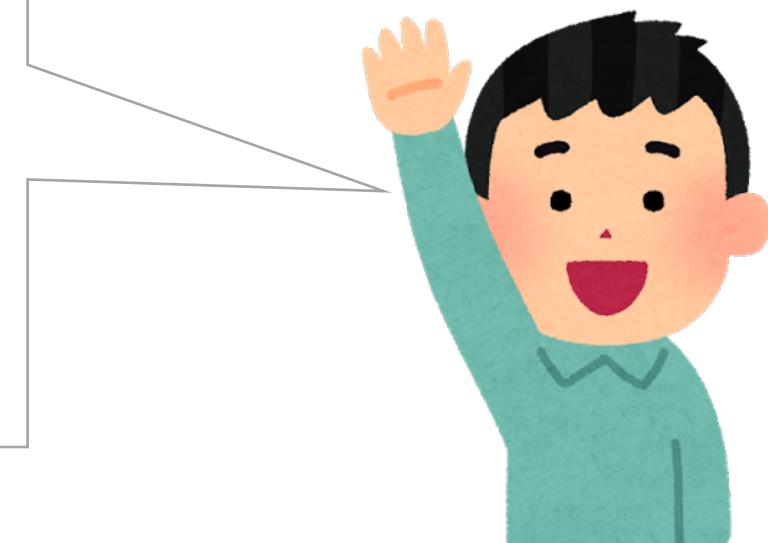
```
# サーバ作成コマンド = server build(実際は1行)
$ usacloud server build
          --os-type centos
          --password [パスワードを入力]
          --name [サーバ名を入力]
```



リソースの一覧表示

```
# サーバー一覧コマンド = server list or ls  
$ usacloud server list # lsでも可
```

先ほど作成した3台が
テーブル形式で表示されるはずです





リソースの一覧表示(オプション)



検索条件やソート条件を
オプション指定してみましょう！

名前で検索

```
$ usacloud server list --name [名前(中間一致)]
```

名前の昇順

```
$ usacloud server list --sort Name
```

IDの降順(降順にする場合、項目名の前に-をつける)

```
$ usacloud server list --sort -ID
```



出力形式の変更

オプションで出力形式を変えられます。

- テーブル(デフォルト)
- CSV/TSV
- JSON
- IDのみ
- カスタム(go言語のtext/template)



```
# 出力形式は--output-type(or --out)等で指定する  
$ usacloud server ls --out [出力形式]
```



出力形式の変更

CSV/TSV

```
$ usacloud server ls --out csv # or tsv
```

CSV/TSV + 列指定(ID列とName列だけ表示)

```
$ usacloud server ls --out csv --col ID --col Name
```

JSON

```
$ usacloud server ls --out json
```

IDのみ(別のコマンドと組み合わせる場合などで利用)

```
$ usacloud server ls --quiet # or -q
```



出力形式の変更(応用)

go言語のtext/template形式も利用できる

```
$ usacloud server ls --format "ID is {{.ID}} Name is {{.Name}}"
```

```
ID is 000000000001 Name is test01
```

```
ID is 000000000002 Name is test02
```

```
ID is 000000000003 Name is test03
```



任意の形式で出力できるため、
色々応用可能です。

例) 監視システム(zabbixなど)との連携



リソースの更新/読み取り/削除

更新

```
$ usacloud server update --tags devel [サーバ名 or ID]
```

読み取り

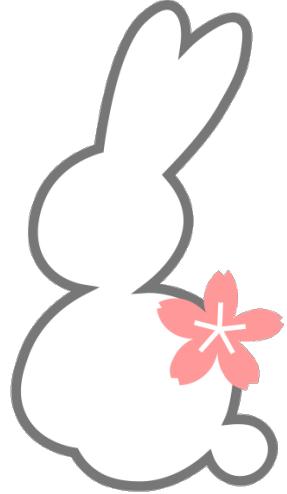
```
$ usacloud server read [サーバ名 or ID]
```

削除(電源OFFも同時にする場合は--forceオプション)

```
$ usacloud server delete [サーバ名 or ID]
```



この後もサーバを使いますので
削除してしまった場合は再作成をお願いします！



usacloud

サーバ操作編



サーバの電源操作

起動

```
$ usacloud server boot [ID or 名前]
```

シャットダウン(ACPI)

```
$ usacloud server shutdown [ID or 名前]
```

強制シャットダウン

```
$ usacloud server shutdown-force [ID or 名前]
```

リセット

```
$ usacloud server reset [ID or 名前]
```



[TIPS] 操作対象の指定方法



IDや名前での指定のほか、
・タグ
・他のコマンドの実行結果
などでも指定可能です。

```
# ID または 名前で指定
```

```
$ usacloud server boot [ID or Name]
```

```
# タグで指定
```

```
$ usacloud server boot --selector [タグ名]
```

```
# 他のコマンドの実行結果で指定(サブシェルの出力など)
```

```
$ usacloud server boot $(usacloud server ls -q)
```



サーバへのSSH接続

```
# SSH(WindowsではSSHクライアントが必要)
```

```
$ usacloud server ssh [ID or 名前]
```

```
# SSHでのコマンド実行(WindowsもOK)
```

```
$ usacloud server ssh-exec [ID or 名前] [コマンド]
```



サーバのIPアドレスや管理ユーザー名を自動設定してくれるよ！
(例：CentOSならrootユーザー、Ubuntuならubuntuユーザーなど)

ssh-execはWindowsでもOK！



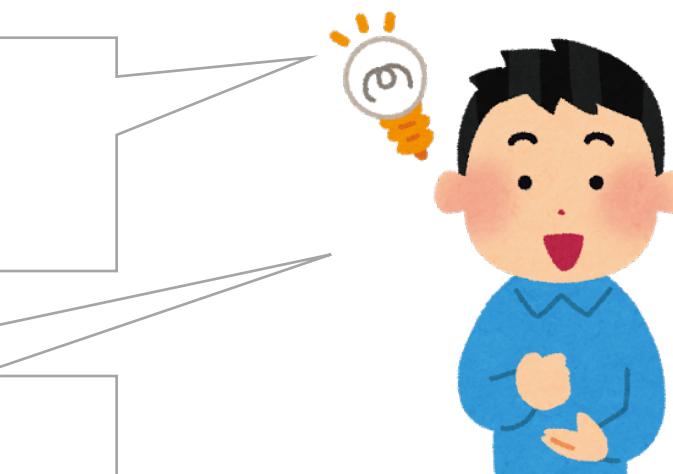
サーバとのファイルのやりとり(SCP)

```
# scpでファイル送信(ローカルからリモート)  
$ usacloud server scp local.txt [ID or 名前]:/root/remote.txt
```

```
# リモートからローカル  
$ usacloud server scp [ID or 名前]:/var/log/syslog ./syslog
```

設定ファイルを送り込んだり、
ログを収集したり、
用途は様々です。

WindowsでもOK！！



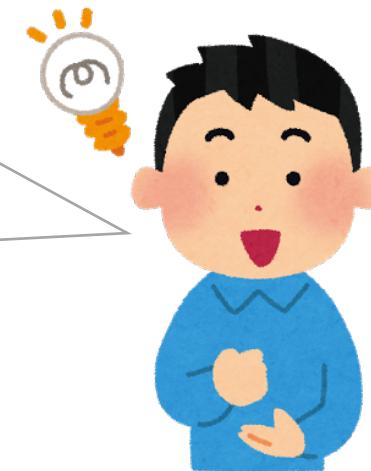


サーバへのVNC接続

```
# VNC(WindowsではVNCクライアントが必要)  
$ usacloud server vnc [ID or 名前]
```

インターネットに接続していない
サーバへもVNC接続できます！

ISOイメージからのOSインストールなどで
効果を発揮！！



※Windowsの場合、拡張子”.vnc”をVNCクライアントに関連付けておく必要があります。





メンテナンス情報の取得

```
# メンテナンスが予定されているサーバの情報を取得  
$ usacloud server maintenance-info
```

※現在のゾーンのサーバが対象



メンテナンス予定がある場合のみ表示されます

zabbixなどの監視システムと組み合わせて
slackなどへ通知されるような仕組みも作れます！

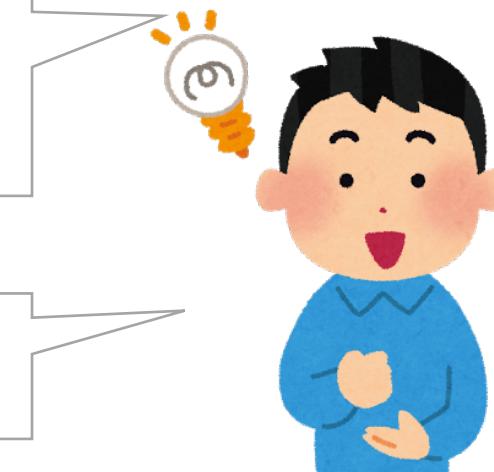


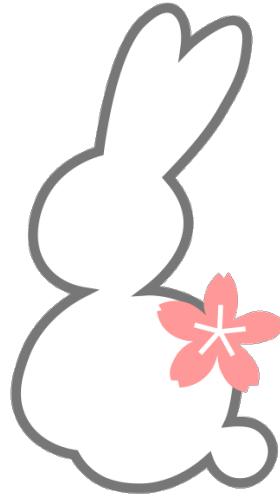
請求CSVのダウンロード

```
# 当月の請求CSVをダウンロード  
$ usacloud bill csv
```

各種フィルタ(tail、cut、awkなど)などと組み合
わせたり、社内システムと連携したり、、、

cronやタスクスケジューラーで定期実行





usacloud

<https://github.com/sacloud/usacloud>

Enjoy!!