

【初心者向け】CentOS 7インスタンスの初期設定手順

#OS #ネットワーク #systemd #CentOS #Amazon EC2 #AWS



虎塚

2015.07.08



68



168



54

こんにちは、虎塚です。

今回は、CentOS 7の公式AMIで頻繁に実施する初期設定の手順をまとめます。CentOS 7からSystemdが導入されたため、CentOS 6までと一部違う部分があります。

設定内容の目次

今回説明するのは、次の内容です。

1. パッケージの更新
2. ロケールとキーマップの変更
3. タイムゾーンの変更
4. 不要なサービスの停止
5. Webサーバのインストールと起動設定
6. OSファイアウォールの設定

「検証用途でちょっとWebサーバを立ち上げたい時」というユースケースを想定して、Webサーバのインストールと起動までを行います。

0. CentOSインスタンスの起動

CentOS wikiに、公式AMIへのリンクがあります。

- [Cloud/AWS - CentOS Wiki](#)

「CentOS-7 x86_64」のURLをクリックして、リージョンに合ったAMIを選択します。今回は、ami-89634988を利用しました。

ここから先の手順では、起動したインスタンスにSSHログインして作業します。なお、**CentOS 6と7のAMIでは、初期ユーザが異なります**ので、気をつけてください。

- CentOS 6: root
- CentOS 7: centos
 - パスワードなしのsudoが可能

1. パッケージの更新

初期インストールされているパッケージを更新します。これは従来と同じですね。

```
$ sudo yum update
```

2. ロケールとキーマップの変更

システムロケールとキーボードレイアウトは、`localectl`コマンドで設定します。まず、現在設定されている値を確認します。

```
$ localectl status
System Locale: LANG=en_US.UTF-8
VC Keymap: us
X11 Layout: us
```

次に、設定可能なロケールを確認します。

```
$ localectl list-locales
aa_DJ
aa_DJ.iso88591
aa_DJ.utf8
[...]
ja_JP.eucjp
ja_JP.ujis
ja_JP.utf8
[...]
```

ロケールの設定を変更します。ここでは、`ja_JP.utf8`に変更しました。

```
$ sudo localectl set-locales LANG=ja_JP.utf8
```

それから、設定可能なキーマップを確認します。

```
$ localectl list-keymaps
ANSI-dvorak
amiga-de
amiga-us
applkey
[...]
jp-dvorak
jp-kana86
jp106
[...]
```

キーマップの設定ファイルを変更します。ここではjp106に変更しました。

```
$ sudo localectl set-keymap jp106
```

最後に、変更後の値を確認します。

```
$ localectl status
System Locale: LANG=ja_JP.utf8
VC Keymap: jp106
X11 Layout: jp
X11 Model: jp106
X11 Options: terminate:ctrl_alt_bksp
```

3. タイムゾーンの変更

タイムゾーンは、timedatectlコマンドで設定します。まず、現在設定されている値を確認します。

```
$ timedatectl status
Local time: 水 2015-07-08 10:32:26 UTC
Universal time: 水 2015-07-08 10:32:26 UTC
Timezone: UTC (UTC, +0000)
NTP enabled: yes
```

```
NTP synchronized: yes
RTC in local TZ: no
DST active: n/a
```

次に、設定可能なタイムゾーンを確認します。

```
$ timedatectl list-timezones
Africa/Abidjan
Africa/Accra
Africa/Addis_Ababa
[...]
Asia/Tokyo
[...]
```

設定を変更します。今回は、Asia/Tokyoに変更します。

```
$ sudo timedatectl set-timezone Asia/Tokyo
```

最後に、変更後の値を確認します。

```
$ timedatectl status
Local time: 水 2015-07-08 19:34:23 JST
Universal time: 水 2015-07-08 10:34:23 UTC
Timezone: Asia/Tokyo (JST, +0900)
NTP enabled: yes
NTP synchronized: yes
RTC in local TZ: no
DST active: n/a
```

設定は即座に反映されます。

4. 不要なサービスの停止

サービスの管理は、systemctlコマンドで行います。まず、サーバ起動時に読み込みが有効になっているサービスを確認します。

```
$ systemctl list-unit-files --type service | grep enabled
auditd.service          enabled
chronyd.service          enabled
```

```
cloud-config.service      enabled
[...]
```

上のサービス一覧から、不要なサービスの起動設定を無効にします。ここでは例として、postfixを停止してみましょう。

次に、停止するサービスの設定状態を確認します。

```
$ systemctl status postfix.service
postfix.service - Postfix Mail Transport Agent
  Loaded: loaded (/usr/lib/systemd/system/postfix.service; enabled)
  Active: active (running) since 水 2015-07-08 15:31:08 JST; 4h 7min ago
 Main PID: 924 (master)
  CGroup: /system.slice/postfix.service
          └─ 924 /usr/libexec/postfix/master -w
          └─ 926 qmgr -l -t unix -u
          └─19126 pickup -l -t unix -u
```

enabledとあるので、サーバ起動時の読み込み有効になっています。また、active (running)とあることから、現在起動中であることが分かります。

対象のサービスを停止し、起動時設定を無効化します。

```
$ sudo systemctl stop postfix.service
$ sudo systemctl disable postfix.service
rm '/etc/systemd/system/multi-user.target.wants/postfix.service'
```

最後に、変更後の状態を確認します。

```
$ systemctl status postfix.service
postfix.service - Postfix Mail Transport Agent
  Loaded: loaded (/usr/lib/systemd/system/postfix.service; disabled)
  Active: inactive (dead)
```

サービスは停止してinactive (dead)状態になり、読み込み設定も無効 (disabled) になりました。

同様の操作を、停止対象とするすべてのサービスに対して行います。

5. Webサーバのインストールと起動設定

今回はApache HTTP Serverを使用します。まず、標準リポジトリからhttpdをインストールします。

```
$ sudo yum install httpd
[...]
```

httpd	x86_64	2.4.6-31.el7.centos	base
-------	--------	---------------------	------

依存性関連でのインストールをします:

apr	x86_64	1.4.8-3.el7	base
apr-util	x86_64	1.5.2-6.el7	base
httpd-tools	x86_64	2.4.6-31.el7.centos	base
mailcap	noarch	2.1.41-2.el7	base

```
[...]
```

次に、インストール直後のサービスの状態を確認します。

```
$ systemctl status httpd.service

httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled)
   Active: inactive (dead)
```

サービスは停止しており、起動時の読み込みも無効になっています。httpdを起動して、OSの起動時にも読み込まれるようにします。

```
$ sudo systemctl start httpd.service
$ sudo systemctl enable httpd.service
ln -s '/usr/lib/systemd/system/httpd.service' '/etc/systemd/system/multi-user.target.wants/httpd.service'
```

最後に、設定後の値を確認します。

```
$ systemctl status httpd.service
httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled)
   Active: active (running) since 水 2015-07-08 20:00:22 JST; 40s ago
   Main PID: 19361 (httpd)
   Status: "Total requests: 0; Current requests/sec: 0; Current traffic: 0 B/s"
   CGroup: /system.slice/httpd.service
           └─19361 /usr/sbin/httpd -DFOREGROUND
           └─19362 /usr/sbin/httpd -DFOREGROUND
[...]
```

ここまでの設定を終えると、インターネットから80番でアクセスできます。ブラウザでhttp://（インスタンスのpublic IPアドレス）にアクセスして、Apache HTTP serverの初期ページが表示されることを確認します。

6. OSのファイアウォールの設定

CentOS 7では、ファイアウォールの設定をfirewalldで行います。AWSを使う際、通常はインスタンスへのアクセス制御をSecurityGroupでおこないますが、場合によってはOSのファイアウォールを有効にしたいことがあるかもしれません。そんなときは、このステップで説明する設定が必要です。

まず、サービスの存在を確認します。

```
$ systemctl status firewalld
firewalld.service
  Loaded: not-found (Reason: No such file or directory)
  Active: inactive (dead)
```

firewalldは初期インストールされていないようです。というわけで、インストールします。

```
$ sudo yum install firewalld
[...]
firewalld                                noarch                0.3.9-11.el7           base
依存性関連でのインストールをします:
ebtables                                x86_64                2.0.10-13.el7          base
python-slip                             noarch                0.4.0-2.el7            base
python-slip-dbus                        noarch                0.4.0-2.el7            base
[...]
```

次に、firewalldの現在の状態を確認します。

```
$ systemctl status firewalld
firewalld.service - firewalld - dynamic firewall daemon
  Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled)
  Active: inactive (dead)
```

サーバ起動時の読み込み設定はあります (enabled) が、サービス自体はまだ起動していません (inactive)。そこで、firewalldを起動します。

```
$ sudo systemctl start firewalld
```

起動後の状態を確認します。

```
$ systemctl status firewalld
firewalld.service - firewalld - dynamic firewall daemon
  Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled)
  Active: active (running) since 水 2015-07-08 20:13:36 JST; 1min 27s ago
  Main PID: 19476 (firewalld)
  CGroup: /system.slice/firewalld.service
          └─19476 /usr/bin/python -Es /usr/sbin/firewalld --nofork --nopid
```

firewalldが起動するとOSファイアウォールが有効になり、ステップ5でインストールしたHTTPサーバにインターネットから繋がらなくなります。 再びアクセスできるようにするために、次の設定を行います。

firewall-cmdコマンドで、現在のファイアウォールの設定を確認します。

```
$ firewall-cmd --list-all
public (default)
  interfaces:
  sources:
  services: dhcpv6-client ssh
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:
```

servicesの行に注目します。ここにhttpがないということは、80番ポートが開いていません。ファイアウォールに80番ポートを開く設定を追加します。

```
$ sudo firewall-cmd --permanent --add-service=http --zone=public
success
```

サーバを再起動しても設定が消えないように、--permanentオプションをつけるのがポイントです。

次のコマンドを実行して、設定を反映します。

```
$ sudo firewall-cmd --reload  
success
```

最後に、設定後の値を確認します。

```
$ firewall-cmd --list-all  
public (default)  
  interfaces:  
  sources:  
  services: dhcpv6-client http ssh  
  ports:  
  masquerade: no  
  forward-ports:  
  icmp-blocks:  
  rich rules:
```

servicesの行にhttpが追加されました。

これで再び、インターネット経由で80番ポートにアクセスできるようになりました。