

パブリッククラウドを利用した技術職員向けサーバ構築講習会の実践報告

古屋保[†], 山之上卓^{†‡}, 下園幸一[†], 小田謙太郎[†], 相羽俊生[†], 湊田孝康[‡], 森邦彦^{†‡}

[†]鹿児島大学 学術情報基盤センター

[‡]鹿児島大学大学院 理工学研究科

furuya@cc.kagoshima-u.ac.jp

概要：我々が所属する鹿児島大学学術情報基盤センター（以下、当センター）では、本学技術職員等のスキルアップとして、Web サイト構築の講習会を毎年度開催している。2011年度までは、受講者持込みのノート型パソコン上でLinux サーバ構築実習を行っていたが、マシンスペックの違いで進行具合に違いが生じるなどのトラブルがあった。2012年度は、パブリッククラウド上で、SSH クライアントソフトウェアを通して、Web アプリケーションの開発実習およびクラウドサービスの利用についての講習を行った。今回は、当センター設置のPC 端末を利用したため、受講者全員が全く同じ環境でトラブルが生じること無く講習を行うことができた。併せて、クラウドサービス API を利用した仮想サーバ（インスタンス）管理ツールを開発準備し、受講者自らが利用するインスタンスを管理することで、クラウド資源（稼働時間等）の節約を実現できた。それらの内容について報告する。

1. はじめに

我々が所属する当センターでは、本学技術職員等のスキルアップの試みとして、Linux を使った Web-DB 連携サーバの構築およびその上で動く Web アプリケーションの基礎的な開発の習得を目的とする Web サイト構築の講習会を毎年度開催している（図 1）。

前年度（2011 年度）までは、受講者持込みのノート型パソコンに LAMP（Linux, Apache, MySQL and PHP）環境を構築し、その上で Web-DB 連携アプリケーションの構築実習を行っていたが、マシンスペックの違いで進行具合に違いが生じるなどのトラブルがあった。そこで、今年度（2012 年度）は、その LAMP 環境をパブリッククラウドである、Amazon Web Services（以下、AWS）の Amazon EC2（以下、EC2）上に構築することにした。これにより、受講者は当センター設置の Windows がセットアップされたパソコンから SSH クライアントソフトウェアを通して、Web アプリケーションの開発実習を行うことができる。しかし、講習会でパブリッククラウドをサーバ構築実習環境として利用するのは、当センターでは初の試みであり、学外サービスを利用することで、色々と不安要素もあった。

本発表では、前年度までの講習内容との違い、および AWS を利用したことによる懸念事項やその解決策等について報告する。

2. 前年度までの講習会内容について

2.1. 講習会の概要

当センターが実施しているサーバ構築講習会は、2008 年度より「Web サイト構築セミナー」というタイトルで開催しており、講習期間は、3 時間 / 日 × 5 日間連続の計 15 時間で、受講料は無料である。当センター教職員および大学院理工学研究科教員である我々が講師として各単元別に担当する。主な内容は以下の通り。

- Linux のセットアップおよび Linux 概要
- Linux システム管理（ネットワーク管理、パッケージ管理、ユーザ管理等）
- Web サーバ構築（Apache）
- データベースサーバ構築（MySQL）
- PHP プログラミングおよび Web-DB 連携プログラミング



図 1 講習会風景

定員は 10 名で、受講対象者は、主に本学教職

員とし、定員を満たさず、かつ希望がある場合は学生も可としている。実際には各部局の技術職員の参加が多い。特に受講要件を設けていないので、受講者スキルは様々である。

使用するテキストは、各担当講師が作成するオリジナル教材で、講習会専用 PukiWiki サイトを準備し、オンラインで利用できるようにしている(図 2)。インターネットに公開しているため、受講者は自宅でも内容を確認することが可能で、予習復習として、また講習会終了後も利用可能である。

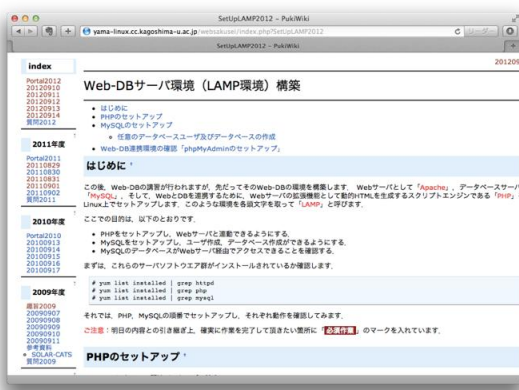


図 2 講習会用テキスト

このテキストにより、受講者各人が Linux によるサーバ環境を一から構築し、最終的にはデータベースと連携した動的 Web アプリケーションの構築と公開 (LAN 内) までを目標とする。

2.2. サーバ構築プラットフォーム

講習会を行う上で、サーバ構築のプラットフォームをどうするかが、いつも問題として挙げられる。開催側である当センターでは、特別に Linux サーバを構築するためのパソコンを準備して揃えることはできないため、受講者がノート型パソコンを持参することにしていく。

開講当初 (2008 年度から 2009 年度) は、受講者が持参したノート型パソコンの内蔵 HDD をフォーマットしてもらい、そこに Linux をセットアップしてもらっていた。当然、それまでパソコンに入っていた環境を壊してしまうことになり、受講者の負担が増える。また、マシンスペックの違い等環境もそれぞれ異なることで個別の対応が必要となり、講師側の負担も増える。

これらの問題があり、2010 年度以降は、各受講者の持込みパソコンに、無償の仮想化ソフトウェアである Oracle VM VirtualBox をセットアップ

し、その上の仮想マシンに Linux をセットアップすることに変更した。これにより、受講者は既存の環境を壊すこと無く、また受講後も自分の構築した Linux サーバ環境を引き続き利用できる。また、構築した Web アプリケーションの実行結果も、仮想的ではあるが別マシン (ホストマシン) の Web ブラウザから参照でき、ネットワークを介すことでのサーバクライアントシステムのユーザ体験ができる。仮想マシンレベルでは同一スペックとなり、講習の進行も以前よりは楽になったが、ホストマシンの物理的なスペックの差が、講習会の進行を左右してしまうこともあった。

3. 今年度の講習会内容について

3.1. サーバ構築プラットフォームの変更

情報システムの構築・運用は、クラウドコンピューティングの市場拡大に伴い、ユーザ自身がサーバを所有し、一からシステムを構築していくという形態から、パブリックに提供されるサーバシステム (インフラ) やサービスを利用する形態に推移しつつある。

当センターにおいても、2009 年度より教育研究用計算機システムに仮想サーバシステムを導入し、部局専用サーバホスティングサービスを展開している。そして、2013 年度より、パブリック VPS サービスを始めることも決定し、また、限定的ではあるが AWS の課金請求代行サービスを先行的に利用する予定である。

これらのことを鑑み、今年度の講習会より、実習で使用する Linux サーバは、パブリッククラウド (AWS) 上で構築することにした。これにより講習を当センターの端末室の Windows パソコンで行うことが可能となり、受講者は自分のパソコンを持参すること無く、また受講者全員の Linux サーバ構築環境を同一とすることができた。

併せて、パブリッククラウドの各種サービスを利用した、新しいスタイルのシステム構築についても講習するために、タイトルも「クラウドコンピューティングセミナー」に変更し、内容を一部変更した (図 3)。

- クラウドサービス (AWS) について、Linux 概要
- Linux システム管理、セキュリティ対策
- Web-DB 開発環境構築
- PHP プログラミングおよび Web-DB 連携プログラミング

● Git 入門, phpfog の利用, SaaS について

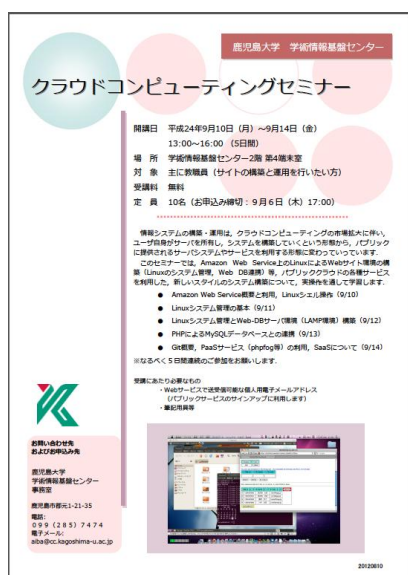


図 3 今回の講習会のパンフレット

3.2. 各種パブリッククラウドサービスの利用について

一般的に、クラウドの構成要素として、IaaS (Infrastructure as a Service), PaaS (Platform as a Service), SaaS (Software as a Service) が挙げられ、今回の講習会では 3 種それぞれのサービスの内容を網羅させた。

IaaS については、AWS の EC2 を利用していることにより対応できる。今回は、全てターミナルクライアントからのオペレーションとなることから、SSH や vi エディタの使い方について重点的に説明し、また、構築するサーバがインターネットに公開されていることからファイアウォール (iptables) の設定方法などセキュリティ対策についても説明した。

PaaS については、PHP+MySQL の実行環境等ミドルウェアレベルまでを無償で提供している「phpfog」(<https://phpfog.com>) を利用することで対応した。実際に、EC2 環境で各受講者が構築した Web アプリケーションを phpfog にデプロイするところまで実習を行った。

SaaS については、Microsoft Office 365 を例に説明することで対応した。本学で既に利用している Microsoft Live@Edu との違い、特にメッセージアプリ「Lync」について実践した。

3.3. AWS の課金について

AWS の各サービスを利用する際は、基本的に有償であり、従量課金制となっている。決済はクレジットカード払いにより行われるため、個人保有のクレジットカード番号を登録しなければサインアップは不可能であり、受講者個人にアカウント登録を強いることはできない。

本来、当センターとして、AWS 決済代行業者と契約し、講習会用の経費で講習会用の AWS アカウントを準備する予定だったが間に合わず、今回は著者個人のアカウントを利用することにした。幸い、2012 年 9 月現在 AWS は新規登録者を対象に、1 年間の無料利用枠を提供しており、EC2 micro インスタンスを 1 ヶ月につき稼働時間 750 時間以内であれば、利用料は無料となっている^[1]。各インスタンスタイプの 1 時間単位の課金 (Linux OS の場合) は、表 1 のとおりである。

講習会期間の 5 日間毎日 24 時間稼働を想定した場合、10 名利用で稼働時間合計 1,200 時間となり、450 時間超で \$12.15 (約 970 円) 課金負担となる。

表 1 AWS EC2 インスタンスタイプ^[2]

タイプ	CPU	メモリ	ストレージ	Linux 料金 (1 時間)
small	1ECU	1.7GB	160GB	\$0.092
medium	2ECU	3.75GB	410GB	\$0.184
large	2ECU	7.5GB	850GB	\$0.368
extra large	4ECU	15GB	1,690GB	\$0.736
micro	Max. 2ECU	613MB	8GB, EBS	\$0.027

※2012 年 9 月現在、アジアパシフィック (東京) リージョンの場合

また、講師 6 名分を準備期間 5 日間合せて計算すると稼働時間合計 2,640 時間となり、時間超過分は 4 千円程になってしまう。しかし、1 日の稼働時間を講習会に必要な 5 時間のみとすると、受講者 5 日間 10 名分で 250 時間、講師分 10 日間 6 名分で 300 時間の合計 550 時間となり、無料利用枠分で十分収まる。従って常時稼働ではなく、インスタンスを適時起動および停止させ、無駄に稼働させない工夫が必要である。

今回、受講希望者は 7 名で、定員は満たすことができなかった。クラウド利用という実験的な要素も含んでいたもので、無理に定員を確保することはしなかった。

3.4. インスタンスの管理について

通常、EC2 インスタンスの起動 (Start) およ

び停止 (Stop) などの管理は、AWS Management Console (以下、AMC) (図 4) で行う。

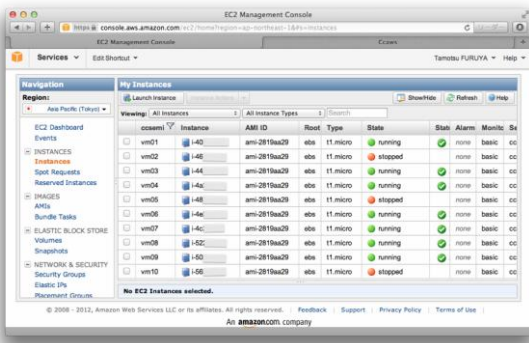


図 4 AWS Management Console 画面

今回使用した micro インスタンスタイプの場合、root デバイスが EBS (Amazon Elastic Block Store) タイプのストレージであるので、インスタンスが停止している間も、内部のデータを保持できる。従って常時稼働させておく必要はない。しかし、外部から参照可能なホスト名 (Public DNS Name) やグローバル IP アドレス (Public IP Address) が、インスタンスの再起動の度に变化してしまう仕様になっているため、その都度、受講者は最新の情報 (ホスト名、IP アドレス) を知る必要があるが、その状態を表示する AMC は、AWS のアカウントを持っている者 (著者) しか利用できない。

4. 受講者用インスタンス管理ツールの開発

4.1. 目的

受講者が利用したいときに、EC2 インスタンスを利用できるようにしなければならないため、受講者自らが、以下のようなインスタンス制御および状態監視ができる管理ツールが必要である。

- 現在のインスタンスの状態監視 (死活監視)
- 現在のインスタンスのホスト名および IP アドレス
- 無駄な課金を抑止するために、インスタンスを適時稼働・停止を行う

4.2. AWS SDK for Ruby^[3]を利用した Ruby on Rails アプリケーション開発

今回、上述した目的を達成するために、我々は AWS の各種サービス API を利用するための開発キットである AWS SDK for Ruby を利用して、独

自のインスタンス管理ツールを開発した。開発キットには、Ruby の他にも Java, PHP などがある中で、今回我々は、環境構築する上で、Gem をインストールするだけの手軽さと、Web アプリケーションフレームワークである Ruby on Rails (以下、Rails) で手軽に利用できることから Ruby を選択した。

Rails で AWS SDK for Ruby を利用するためには、図 5 に示すような Gemfile を準備し、bundle install を実行するだけで環境が整う。

```
source "https://rubygems.org"
gem "rails"
gem "aws-sdk"
```

図 5 Gemfile の内容

AWS はアカウントを識別するために、アクセスキー ID とシークレットアクセスキーで検証する。アプリケーションの初期化スクリプトとして、図 6 に示すようなメソッドを用いて、コンフィグレーションを行う。^[4]

```
AWS.config ({
  :access_key_id => 'アクセスキーID',
  :secret_access_key => 'シークレットアクセスキー',
})
```

図 6 AWS の初期化スクリプトの内容

コンフィグレーションによって AWS へのアクセスが可能になれば、EC2 クラスのオブジェクトを生成し、EC2 インスタンスを識別するインスタンス ID を指定することで、そのインスタンスのオブジェクトを取得することができる (図 7)。

```
ec2 = AWS::EC2.new
instance = ec2.instances[instance_id]
```

図 7 EC2 インスタンスの取得

後は、取得したインスタンスオブジェクトの start/stop メソッドで、インスタンスの起動/停止が制御でき、status メソッドで、現在のインスタンスの状態 (稼働中、停止中等)、そして、public_dns_name メソッドでインスタンスのホスト名が取得できる。^[5]

オブジェクトさえ取得できれば、後は、Rails の機能を利用して、インスタンスの状態監視および制御が HTTP プロトコルにより、Web ブラウザ上で行えるように開発する。

予め、受講者および担当講師分の数の EC2 イン

スタンスを AMC 上で作成した。この作業は、インスタンスタイプとその生成数を指定するだけである。インスタンスごとにタグ名を付け、利用者には、タグ名で自分のインスタンスを識別させるようにした。

4.3. 実行結果

作成した Web アプリケーションは、開発マシン上で動作確認し、Apache+Passenger 構成で、公開サーバにデプロイした。

作成した Web アプリケーションの実行画面を、図 8 に示す。これにアクセスする利用者は、常に、インスタンスの死活監視と、最新のホスト名および IP アドレスを確認でき、またインスタンスの起動と停止を制御できる。

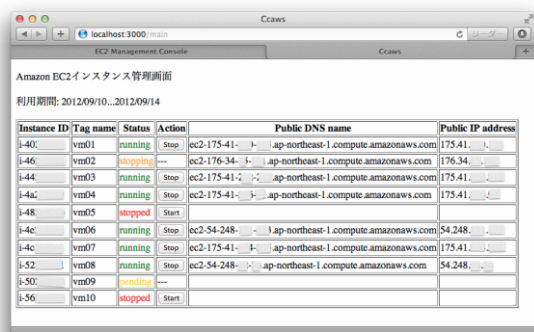


図 8 作成した Web アプリケーション実行画面

認証については、短期の利用であり、共通アカウントでの利用で済むので、簡易に Basic 認証を用いた。

5. 講習結果

5.1. 受講者アンケート

実習の効果を測定するために、受講者に対し最終日に、図 9 のような内容のアンケートを行った。

- 質問 1. 講習会が開催されることを何で知りましたか
- 質問 2. 講習期間 (5 日間という期間) について
- 質問 3. 講習時間 (3 時間/日という時間) について
- 質問 4. 開催時期について
- 質問 5. セミナーの進行について
- 質問 6. 事前知識について
- 質問 7. セミナーの内容レベルについて
- 質問 8. 教材、資料について
- 質問 9. 講習環境について
- 質問 10. もっとより深く学習したい項目について
- 質問 11. 有用性が感じられたものを挙げてください
- 質問 12. 今後、どのような講座を希望しますか?
- 質問 13. セミナー講習会全体について
- 質問 14. ご意見・ご感想をご自由にご記入ください

図 9 アンケートの内容

受講者 7 名のうち、欠席した者を除いた 6 名 (教員 1 名、技術職員 5 名) から回答を得た。

質問 2 と質問 3 の講習時間に関する回答結果を図 10 に示す。期間中は、学生は夏休みということもあり、月曜から金曜の 5 日間という期間であっても、受講者の業務への影響は少なかったと思われる。また、1 日 3 時間の講習時間は適当だったようだ。

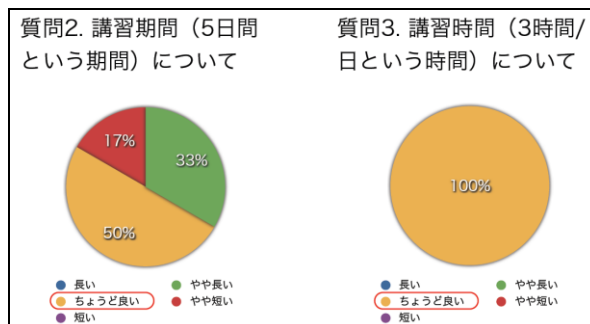


図 10 質問 2 と質問 3 の回答結果

質問 6 と質問 7 の講習レベルに関する回答結果を図 11 に示す。日常からシステム管理されている技術職員は事前知識をある程度持たれていたため、このような結果になっていると推測できる。

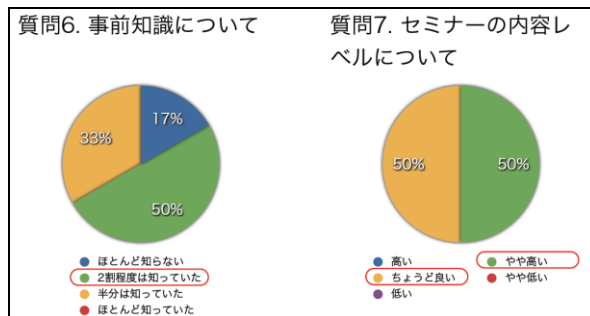


図 11 質問 6 と質問 7 の回答結果

質問 9 と質問 13 についての回答結果を図 12 に示す。講習環境については、1 名の人が「やや不満」と回答しているが、講習環境および講習会全体を通して、ほぼ満足という結果を得た。

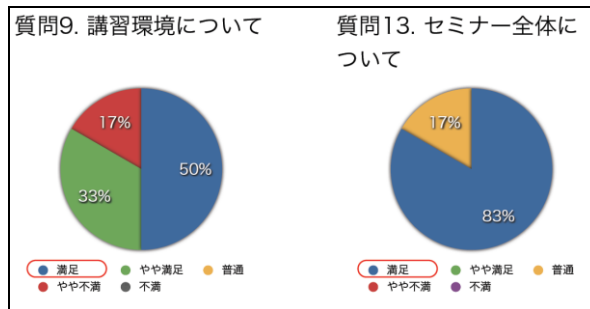


図 12 質問 9 と質問 13 の回答結果

質問 12 についての回答結果を図 13 に示す。今

回、Webアプリケーション開発（プログラミング）についての内容は1日だけであった。受講者はそれ以上の時間を使って、もう少し詳しく講習して欲しかったと見受けられる。

- 今回のセミナーでやった PHP をもう少しやってみたいです。
- 情報処理技術者応用以上の午後試験に対応した講座。
- PHP の概要と MySQL データベースとの連携の講座が途中混乱してしまったので、もう少し詳細にあればいいと思います。
- もう少し優しく、多くの方がきて勉強してほしいと思います。

図 13 質問 12 の回答結果（抜粋）

質問 14 についての回答結果を図 14 に示す。最初のディスプレイ環境に関する意見は、質問 9（図 12）で講習環境に「やや不満」と回答された方の意見である。使用した端末室の液晶ディスプレイは、15.6 型ワイドパネル（解像度 1,360×768 ドット）であり、テキストもオンラインで提示しているの、いくつかウィンドウを開いていなければならない、作業領域としては狭かったものと思われる。

- とても有意義な1週間でした。ありがとうございました。ウィンドウを複数開いて作業したので、ディスプレイを2つ以上、もしくはもう少し（かなり）大きめのディスプレイがあると良いと思いました。
- 興味深く聞きました。ありがとうございました。これから安全にクラウドを利用する方法を逐次セミナーを開いてほしい。
- とても勉強になりました。このセミナーでプログラムに触れたことでもう少し積極的にプログラムに触ってみようと思いました。コンピュータへの興味も増えたので、またセミナーが開催された時には参加します。5日間ありがとうございました。

図 14 質問 14 の回答結果（抜粋）

6. おわりに

パブリッククラウドサービスを利用した講習会ということで、ネットワークやクラウドサービス自体がダウンしてしまうのではないかなど、色々と不安要素があった。しかしながら、受講者の評価も概ね満足との回答を得られ、特段のトラブルも無く円滑に講習会を終えることができた。また、既に用意されたインフラを利用できることで準備が楽であったことや、同一の環境でシステム構築実習できることにより、講師側にとっても大変講習しやすい環境であった。

懸念していた AWS の利用コストについても、利用者が個々でインスタンスを管理制御できるツールを開発し準備したことで、インスタンスの無

駄な稼働時間を抑制し、最終的に全インスタンスの総稼働合計時間を、AWS の無料利用枠内の 750 時間以内に収めることができた。

AWS を利用することのメリットとして、スナップショットによるイメージのバックアップやリストアも可能で、構築するシステムに不具合が起きてしまった場合にも対応でき、また、図 15 に示すように、各インスタンスの CPU の使用率や、ネットワークにおけるデータの入出力などをモニタすることができるなどが挙げられる。

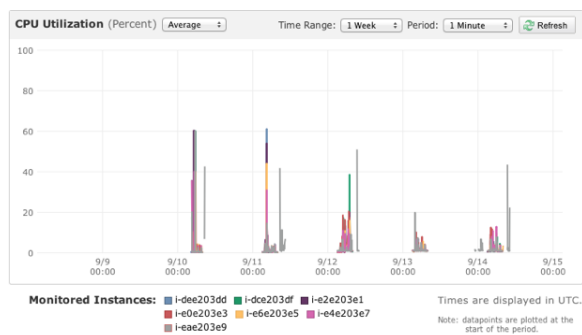


図 15 CPU 平均使用率（講習期間中）

来年度は、インフラおよびシステム管理関係と、プログラム開発とに分けて講習会を開催することで、より受講希望者のニーズに対応したものとなるよう計画する予定である。

参考文献

- [1] 「AWS クラウドサービス無料 利用枠のご案内」, <http://aws.amazon.com/jp/free/>
- [2] 「Amazon EC2 料金表」, <http://aws.amazon.com/jp/ec2/pricing/>
- [3] 「AWS SDK for Ruby」, <http://aws.amazon.com/jp/sdkforruby/>
- [4] 「Getting Started with the AWS SDK for Ruby : Articles & Tutorials : Amazon Web Services」, <http://aws.amazon.com/articles/8621639827664165>
- [5] 「AWS SDK for Ruby API リファレンス」, <http://docs.amazonwebservices.com/AWSRubySDK/latest/frames.html>