目次構成

2016/01/19

川口貴大

1. 研究背景
   1. IaaSの動向
   2. OpenStackの概要
   3. オーケストレーションの定義

本実験では、オーケストレーションとは「構成要素として単一のサーバーのみを定義するだけでなく、複数の構成要素に対して構築を自動化しサポートする」と定義する。

* 1. Heatの概要
  2. 現状の問題点
* Heatテンプレートファイルの複雑な書式
  + 何を入力すればいいのかわかりにくい
  + インデントの深さでテンプレートファイル内での入力区別をしている
* テキスト記述量
  + 項目をひとつ追加するたびに、関連する内容を全て追加で入力しなければならない
* テンプレートファイルから構成情報を把握することの難しさ
  + 膨大な量の文字が入力されており、一見して構成を把握しにくい。
  1. 問題点の解決方法
* Heat専門知識の排除
  + 入力者側が細かな書式を気にしないで済むようなもの
* 入力内容の明確化
  + 何を入力すればよいのか項目名を追加
* テキスト記述量の削減
  + インスタンス名記述項目以外の手動入力を撤廃、プルダウンメニューによる選択肢を提供する。
* 構成情報の可視化
  + 現在構築している構成情報についてアイコンを用いて可視化。

1. GUIエディタの提案
   1. GUIエディタの概要
   2. GUIエディタの要件

* OpenStackに関する基本的な知識は必要
  + Heatを使用するため、最低限基本的な知識は必要
* Heatに関する専門的な知識を有していなくても使用可能
  + 問題点の解決方法である「Heat専門知識の排除」に該当。事前の専門知識に関する学習時間を排除することでスムーズにHeatテンプレートファイルを作成できるようになる。
* 入力内容が明確化されておりテキスト記述量を少なくする
  + 問題点の解決方法、「入力内容の明確化」と「テキスト記述量の削減」に該当。記述量を少なくすることでテンプレートファイル作成にかかる時間を大幅に短縮し、テンプレートファイル作成時に発生するエラーを発生しないようにする。
* アイコンを用いて現在構築している構成を可視化する
  + 構築途中にどのような構成になっているのか確認できる。
  1. Heatで扱うリソース
* ネットワーク
* サブネット
* ルーター
* ルーターインターフェイス
* インスタンス
  1. リソースの依存関係
  2. テンプレートファイルへの出力補助方法

従来方式で繰り返し入力しなければならなかった記述を全てGUIエディタ側で自動入力させる。繰り返しの入力が必要な部分であれば必要数繰り返し入力をさせ、GUIエディタ使用者にはテンプレートファイルを構築するたびに入力内容が変わる部分のみを入力してもらう。

* 1. 入力されたデータの扱い

詳細入力画面で入力されたデータは、データ保管用のクラスへ保管される。その後テンプレートファイルへ出力する時にデータを呼び出し、入力する。

1. GUIエディタの実装
   1. 動作環境
   2. 画面構成

(実際のGUIエディタ画面を貼り付けておく。)

* + 1. 構成確認画面

構成確認画面では、テンプレートファイルで構築する構成が現在どうなっているのか確認することができる。また、この画面上に表示されているインスタンスを表すアイコンをクリックすることで後述する「詳細入力画面」へ遷移できる。全ての入力が終了した後、テンプレートファイル出力ボタンをクリックすることでテンプレートファイルが出力される。

* + 1. 詳細入力画面

詳細入力画面では、インスタンスに関する項目を入力する。入力する項目は「インスタンス名」、「使用するイメージ」、「インスタンスのフレーバーサイズ」、「接続先ネットワーク（ルーター）」の４つとなる。この中でインスタンス名のみ手動入力が必要だが、その他の項目は全てプルダウンメニュー形式で選択肢を選択することで入力する。

* 1. テンプレートファイル出力の流れ
     1. インスタンスに関する記述について

作り出すインスタンスに関する記述は詳細入力画面を用いて入力する。基本的には設定項目をクリックすることで複数の選択肢を表示、選択させることで記入させる。唯一インスタンス名のみ手動で入力してもらう。

* + 1. ネットワークに関する記述について

本来heatでは、ルーターをひとつ増やすごとにルーターに関係する記述を全て追加で入力させなければならない。GUIエディタ上では、ルーター追加ボタンをクリックすることでルーターに関係する記述は利用者に一切入力させることなく新たなルーターと、ルーター内部のアドレス範囲指定を自動でテンプレートファイルに入力する。また、ルーターを増加させるとインスタンス側から接続先として参照できるようにもなる。

1. 評価
   1. 評価の目的

GUIエディタを使用してテンプレートファイルを作成した場合、従来方式である手動でテンプレートファイルを作成した場合それぞれにかかる記述時間と記述内容ミスを比較。GUIエディタを使い従来方式で問題となっていた問題点を解決できたことを証明する。

* 1. 評価内容

初めにGUIエディタ、従来方式それぞれについて記述方法や使用方法を用意したドキュメントを配布し学習してもらう。学習時間を記録する。その後GUIエディタ、従来方式それぞれ７通りの構成についてheatテンプレートファイルを作成してもらい、「作成したテンプレートファイルが正常に動作したと確認できるまでの時間」とその間に発生した記述エラーの数を記録する。

* 1. 評価環境
  2. 結果
  3. 考察

1. 結論
   1. 研究のまとめ
   2. 今後の課題