Automating Infrastructure Provisioning with Chef and Cloud Providers

Table of Contents

Description

Chef's integration with cloud providers like **AWS**, **Azure**, and **Google Cloud Platform (GCP)** enables you to automate the provisioning and management of cloud resources. By defining infrastructure as code within Chef, you can create, configure, and manage cloud infrastructure directly from your Chef Workstation.

Problem Statement

Manually managing cloud resources can lead to inconsistencies and errors. Automating infrastructure provisioning with Chef allows for:

- Consistent and repeatable infrastructure setups.
- Simplified scaling and management of cloud resources.
- Improved control over infrastructure through versioned Chef recipes.

Prerequisites

Completion of all previous lab guides (up to Lab Guide-07) is required before proceeding with Lab Guide-08.

Software Required

- **Chef Workstation**: For creating and running provisioning recipes.
- **Cloud SDKs/CLI**: Required for the cloud provider being used (e.g., AWS CLI for AWS).
- **Chef Plugins**: Chef-Provisioning and cloud-specific drivers.

Hardware Requirement

- A **Chef Workstation** with internet access to connect to the cloud provider API and provision resources.

Implementation Steps

Step-1: Install Required Chef Plugins

Chef requires **Chef-Provisioning** and specific cloud drivers to manage cloud resources. Install these on your Chef Workstation:

1. **Install Chef-Provisioning**:

```
chef gem install chef-provisioning
```

2. **Install Cloud Provider Drivers**:

   - For AWS:

```
chef gem install chef-provisioning-aws
```

   - For Azure:

```
chef gem install chef-provisioning-azurerm
```

   - For Google Cloud:

```
chef gem install chef-provisioning-gce
```

Step-2: Configure Cloud Provider Access

1. **Set up Credentials** for your cloud provider. Ensure that your workstation has the appropriate environment variables or credential files for the provider you are using.

   - **AWS**: Set up `~/.aws/credentials` file or configure environment variables.
   - **Azure**: Use the Azure CLI to log in.
   - **Google Cloud**: Use `gcloud auth login` to configure access.

2. **Verify Access**: Run the relevant CLI commands to verify connectivity to the cloud provider:

   - AWS: `aws ec2 describe-instances`
   - Azure: `az account show`
   - Google Cloud: `gcloud compute instances list`

Step-3: Define Infrastructure as Code with Chef Provisioning Recipes

Chef Provisioning recipes describe the cloud resources you want to create. Here's an example of a Chef recipe to provision an **EC2 instance** on AWS.

Example Recipe: `aws_provision.rb`

```ruby
require 'chef/provisioning/aws_driver'

with_driver 'aws::us-west-2'

machine 'my-ec2-instance' do
  machine_options({
    bootstrap_options: {
      instance_type: 't2.micro',
      image_id: 'ami-0abcdef1234567890',  # Replace with a valid AMI ID
      key_name: 'my-key-pair'
    },
    ssh_username: 'ec2-user'
  })
  action :converge
end
```

In this example:

- `with_driver 'aws::us-west-2'`: Specifies the AWS region.
- `machine 'my-ec2-instance'`: Defines the EC2 instance resource with a specific instance type and AMI ID.
- `action :converge`: Ensures the instance is created if it doesn't already exist.

Step-4: Provision Infrastructure with Chef

1. **Run the Chef Recipe**: Run the recipe to provision the instance:

   ```
   chef-client -z aws_provision.rb
   ```

2. **Verify the Instance**: Check your cloud provider's dashboard or use the CLI to verify that the instance is up and running.

   - AWS: `aws ec2 describe-instances --filters "Name=tag:Name,Values=my-ec2-instance"`
   - Azure: `az vm list --query "[?name=='my-ec2-instance']"`
   - Google Cloud: `gcloud compute instances list --filter="name:my-ec2-instance"`

3. **Apply Additional Configuration**: Use Chef recipes to install software, configure settings, and deploy applications on the provisioned resources.

References

- Chef Documentation
- Chef-Provisioning GitHub
- AWS CLI Documentation

- [Azure CLI Documentation](#)
- [Google Cloud CLI Documentation](#)