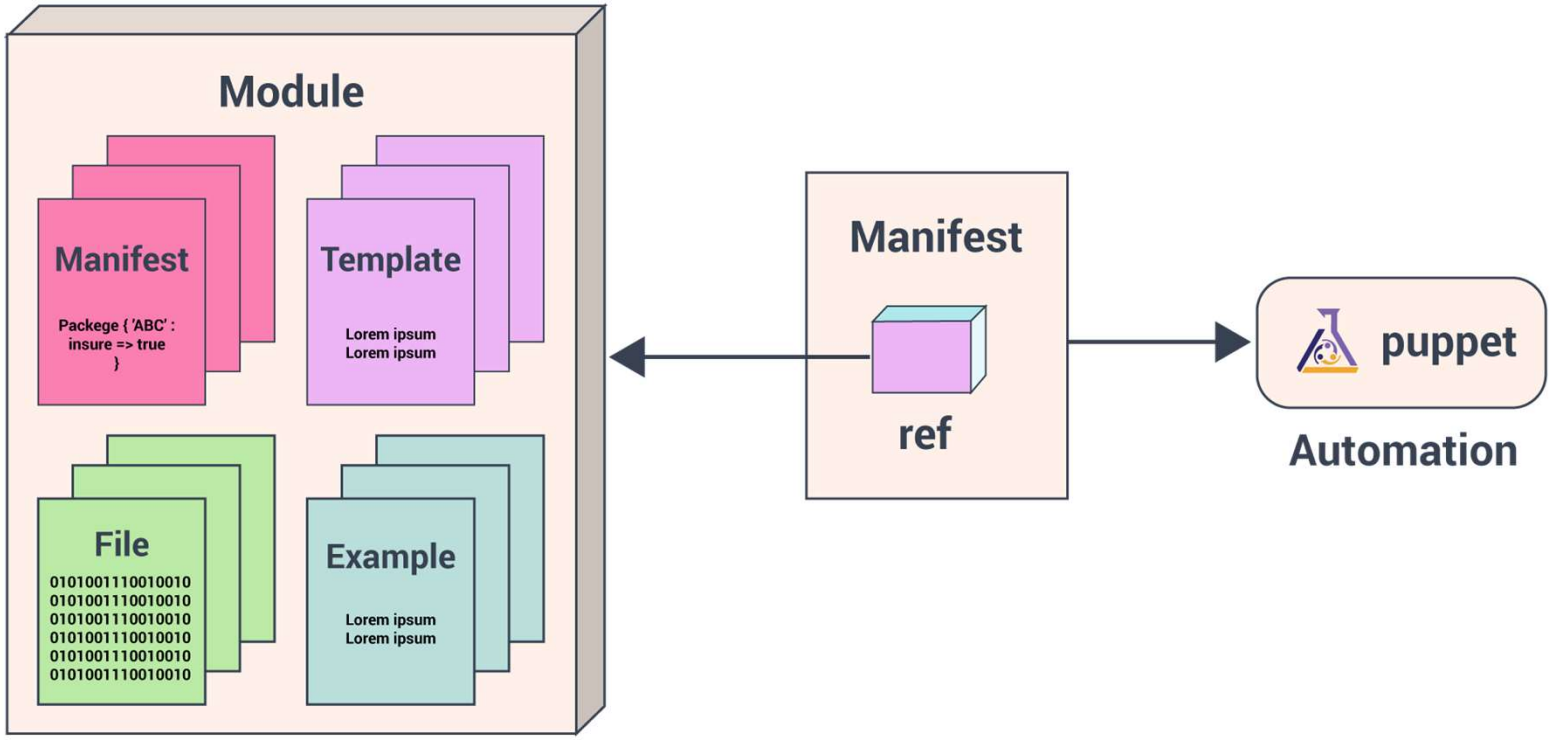
# Introduction to Puppet Programming

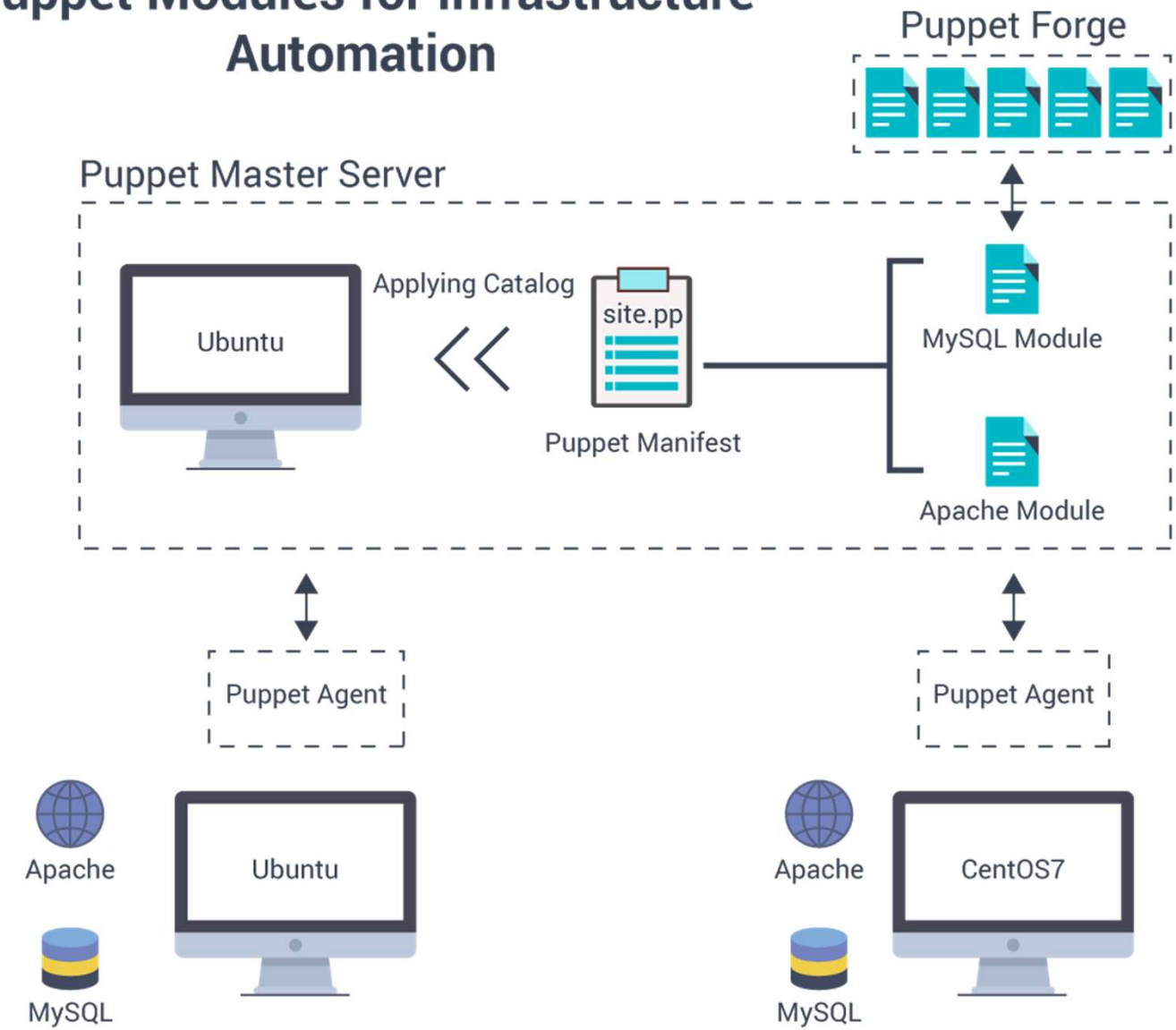
# Key terms in Puppet Programming

- Manifests
- Classes
- Resources
- Puppet Modules

# Puppet Modules



# Puppet Modules for Infrastructure Automation



# Hiera

# What is Hiera

- Configuration software from Puppet labs
- Separates configuration code from functionality
- Introduced in 2011
- Evolved from a simple plugin
- Is now a part of core Puppet

# What is Hiera

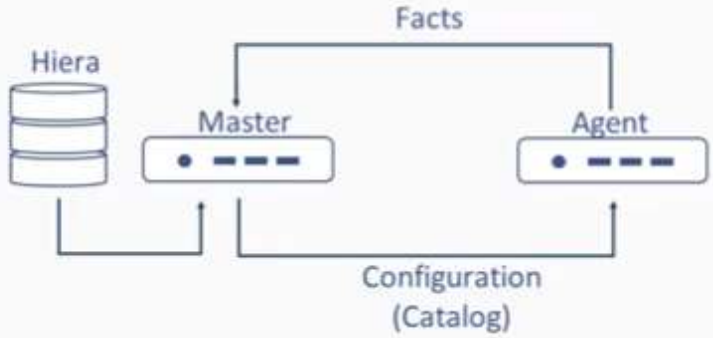
- Built in key value configuration data lookup store.
- Powerful way to store (class parameter) data outside of your pp files
- Stores this data in a efficient hierarchial structure so to minimize code duplication.
- Useful when
  - You want to declare a class that requires a lot of class parameters

# Puppet without Hiera

- Puppet agent asks for the configuration settings
- The master 'downloads' a configuration catalog to the agent

# Puppet with Hiera

- Agent will send facts to the master
- Master will determine the agent by the facts and push the configuration (catalog)
- Hiera has all the value pair configurations in a database
- The configurations are stored in a YAML or JSON file



# Pros

- Pros:
  - Separation between data and code
  - Clandestine storage
  - Integrates with back-end datastores
  - Has conditional logic

# Cons

- Cons:
  - Can be confusing
  - YAML is bad
  - Hard to debug

# Hiera configuration file

- Hiera has it’s own config file, called hiera.yaml.
- Resides in “/etc/puppetlabs/code” directory
- It is yaml file named “hiera.yml”

# hiera.yaml

- Before you can start using hiera, you first need to tell hiera (via the hiera.yaml file):
  - What types of files to search through (i.e. yaml files, or json files, or both)
    - We specify this under the “:backend” setting.
  - What are the names of these data files
    - We specify this under the :hierarchy setting.
  - In what order to look look through these files
    - We specify this under the “:hierarchy” setting
    - Also in the “:backend” setting which dictates which file type should be scanned first
  - In which directory all the data files are located
    - We specify this under the “:datadir” setting.

# hiera.yaml

- All this information is provided to Hiera via the hiera.yaml file:
  - # rpm -qc hiera /etc/hiera.yaml
  - # cat /etc/hiera.yaml
    - ---
    - :backends:
    - - yaml
    - - json

    - :hierarchy:
    - - fileA
    - - fileB
    - - global

    - :yaml:
    - :datadir: /etc/hieradata/yaml

    - :json:
    - :datadir: /etc/hieradata/json

# Create file

- vim /etc/hieradata/yaml/global.yaml
  - ---
  - dad: homer

- hiera dad
  - homer

# A sample YAML based configuration

```
---
ldap_servers:
  - 10.132.17.196
  - 10.132.17.195

users:
  joe:
    home: '/home/joe'
  jenkins:
    password: 'mysecret'
```

# Accessing Hiera Data using CLI

- hiera ldaps_ervers

- If you have used interpolation in the “:datadir” configuration, You should add the parameters as shown below.
  - hiera ldap_servers ::environment=production

# Accessing Hiera Data From Modules

- Use the following syntax in your module to access the data directly.
  - $ldapservers = hiera("ldap_servers")

- $ldapserver is just a puppet variable.
- You can substitute hiera without assigning it to a variable.

- Can set a default value
  - $ldapservers = hiera_array("ldap_servers","10.32.34.45")

# Thanks