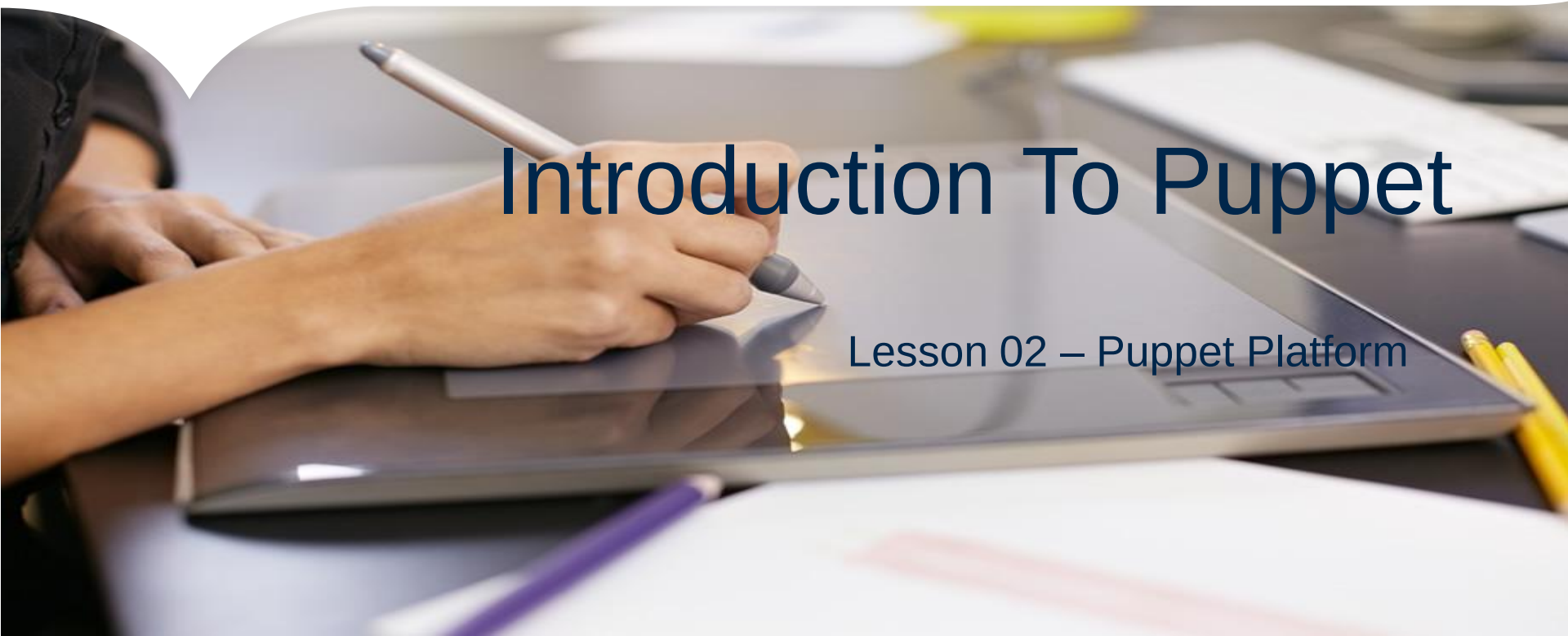
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING



# Introduction To Puppet

## Lesson 02 – Puppet Platform

# Lesson Objectives

- Components of Puppet
- Architecture of Puppet Platform
- Agent/Master Architecture
  - Workflow
  - Data types
  - Components
- Stand-alone Architecture

# Components of Puppet

- Puppet language
  - The language alone is not the full Puppet solution
- Puppet Platform that manages infrastructure
  - Deployment of Puppet language code across the infrastructure
  - Periodically update code with configuration changes
  - Fixing the issues arising due to unintended changes
  - Monitoring systems to ensure everything is working as expected
- Puppet Platform Options
  - Open Source
  - Puppet Enterprise

# Architecture of Puppet Platform

- Puppet platform supports two architectures
  - **Agent/Master architecture**
  - **Standalone architecture**

# Agent/Master Architecture

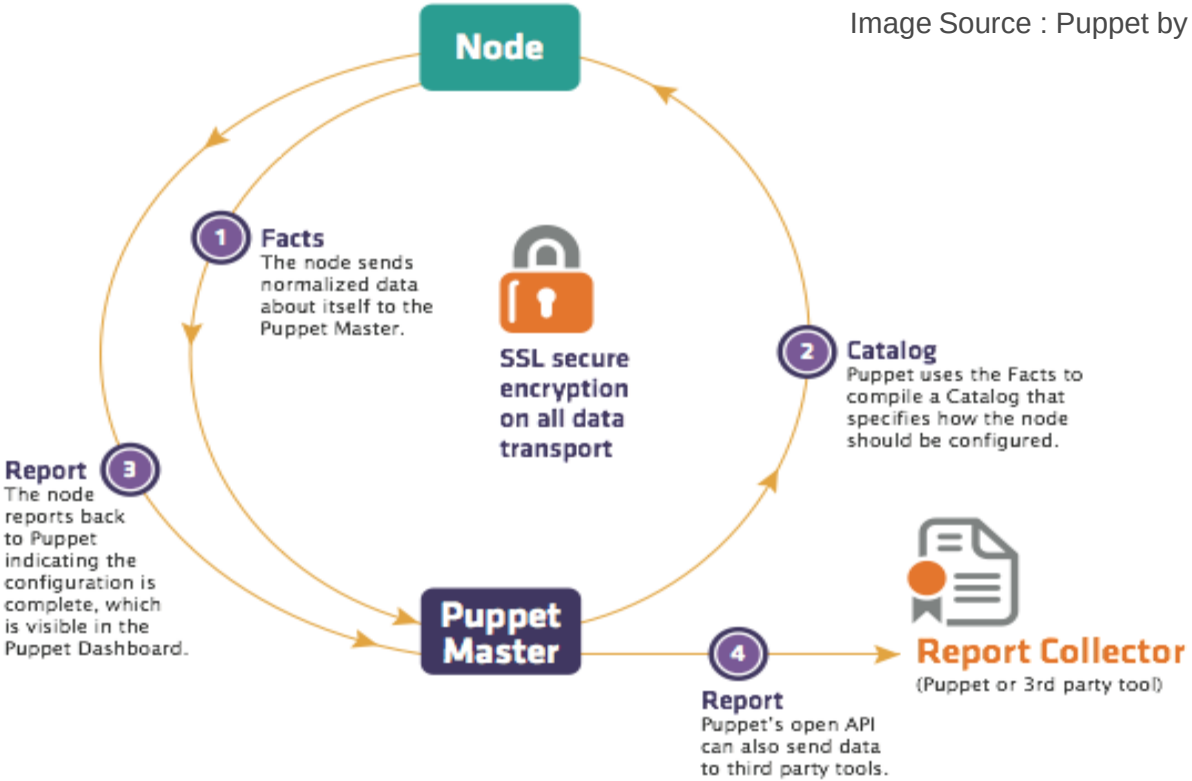
- Puppet usually runs in an agent/master architecture
  - Managed nodes run the Puppet Agent application, usually as a background service
  - One or more servers run the Puppet master application, in the form of Puppet Server
- Periodically, Puppet agent sends *facts* to the Puppet master and request a catalog
- The master compiles and returns that node's *catalog*, using several sources of information it has access to
- Once it receives a catalog, Puppet agent applies it by checking each *resource* the catalog describes

# Agent/Master Architecture <continued …>

- If the agent finds any resources that are not in their desired state, it makes any changes necessary to correct them
- After applying the catalog, the agent submits a report to the Puppet master
- The server has copies of all the data as well as all of the Puppet modules, along with any back-end databases and services that might be needed to compile the configuration

# Workflow of Agent/Master Architecture

Image Source : Puppet by Luke Kanies



# Important data types of Puppet

- **Facts:** System data collected on each machine and used to compile configurations
- **Manifest:** Files containing Puppet code, generally organized into collections called "modules"
- **Catalog:** A graph of a given host's resources to be managed and the dependencies between them
- **Report:** The collection of all events generated during application of a given Catalog

# Components of Puppet Architecture

- **Agent**
  - This component in a Puppet run is the agent process
  - The agent has little functionality of its own; it is primarily configuration and code that implements the client-side aspects of the workflow

- **Facter**
  - This component is an external tool called Facter
  - Used to discover information about the host it is running on
  - Thie information like the operating system, IP address, and host name etc.
  - Facter is extensible so many organizations add their own plugins to discover custom data
  - The agent sends the data discovered by Facter to the server, at which point it takes over the workflow

# Components of Puppet Architecture

- **External Node Classifier (ENC)**
  - Accepts the host name and returns a simple data structure containing the high-level configuration for that host
  - Runs as a separate service or application
  - Specifies functional classes a given host belongs to, and parameters to be used for configuring those classes

- **Compiler**
  - Puppet has a custom language built for specifying system configurations which has its compiler

# Components of Puppet Architecture

- **Transaction**
  - Transactions run on the client
  - It pulls the Catalog down from server
  - Puppet transactions are not atomic
- **Resource Abstraction Layer**
  - It defines resources and how resources can get work done on the system
  - Puppet's language is built to specify resources as modeled by the RAL
- **Reporting**
  - It is the process of generating report
  - Report consists of the events generated by changes to the system.

# Stand-alone Architecture

- Each managed server has its own complete copy of your configuration info and compiles its own catalog
- Stand-alone architecture can be used in what amounts to a client/server configuration, by pulling all configuration files to each client and having it parse them directly

# Lesson Summary

- Components of Puppet
- Architecture of Puppet Platform
- Agent/Master Architecture
  - Workflow
  - Data types
  - Components
- Stand-alone Architecture