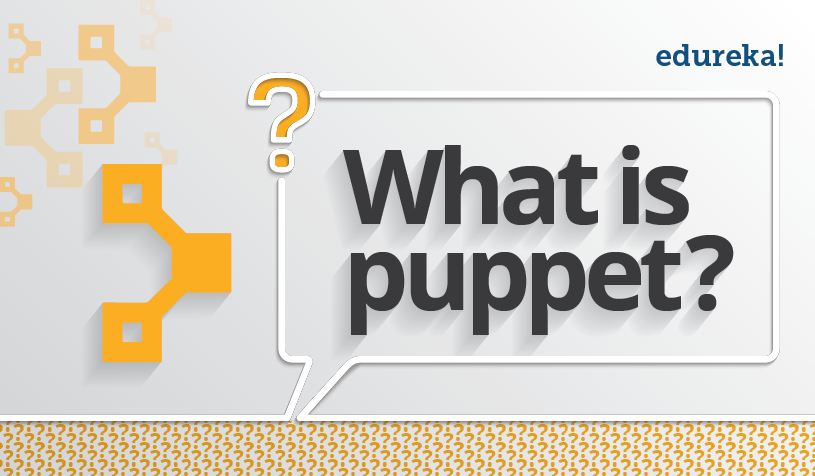
What is Puppet ? – Configuration Management Using Puppet

[Saurabh](https://www.edureka.co/blog/author/saurabhedureka-co/) Nov 18, 2016 1,174 [**0**](https://www.edureka.co/blog/what-is-puppet/#disqus_thread)

Today, the most mature tool for Configuration Management is Puppet. But, I know you must be wondering why is Puppet so popular and what makes it unique, when compared to other Configuration Management tools. In this “**What is Puppet**” blog, I will answer these questions for you.

**What Is Puppet?**

Puppet is a Configuration Management tool that is used for deploying, configuring and managing servers. It performs the following functions:

* Defining distinct configurations for each and every host, and continuously checking and confirming whether the required configuration is in place and is not altered (if altered Puppet will revert back to the required configuration) on the host.
* Dynamic scaling-up and scaling-down of machines.
* Providing control over all your configured machines, so a centralized (master-server or repo-based) change gets propagated to all, automatically.

Puppet uses a Master Slave architecture in which the Master and Slave communicate through a secure encrypted channel with the help of SSL. Puppet architecture is explained in detail in my [***Puppet Tutorial***](http://www.edureka.co/blog/puppet-tutorial/?utm_source=blog&utm_medium=content-link&utm_campaign=what-is-puppet)blog.

Now that you know what is Puppet, let us understand the reasons behind the popularity of Puppet.

**What Is Puppet – Key Metrics**

Below are few facts about Puppet:

* **Large installed base:** Puppet is used by more than 30,000 companies worldwide including Google, Red Hat, Siemens, etc. along with several universities like Stanford and Harvard law school. An average of 22 new organizations per day use Puppet for the first time.
* **Large developer base:** Puppet is so widely used that lots of people develop for it. Puppet has many contributors to its core source code.
* **Long commercial track record:**Puppet has been in commercial use since 2005, and has been continually refined and improved. It has been deployed in very large infrastructures (5,000+ machines) and the performance and scalability lessons learned from these projects have contributed in Puppet’s development.
* **Documentation:**Puppet has a large user-maintained wiki with hundreds of pages of documentation and comprehensive references for both the language and its resource types. In addition, it’s actively discussed on several mailing lists and has a very popular IRC channel, so whatever your Puppet problem, it’s easy to find the answer.
* **Platform support:** Puppet Server can run on any platform that supports ruby for ex: CentOS, Microsoft Windows Server, Oracle Enterprise Linux etc. It not only supports the new operating systems but it can also run on relatively old and out-of-date OS and Ruby versions as well.

It is now evident that Puppet has huge demand globally. But, before diving deep into Puppet, it’s only fair that I first explain what is Configuration Management and why it is important.

**Configuration Management**

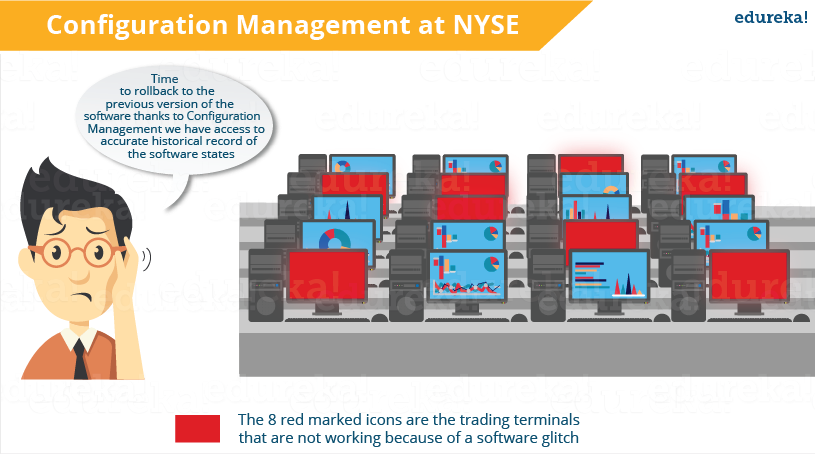
System Administrators usually perform repetitive tasks such as installing servers, configuring those servers, etc. They can automate this task, by writing scripts, but it is a very hectic job when you are working on a large infrastructure.

To solve this problem, *Configuration Management* was introduced. Configuration Management is the practice of handling changes systematically so that a system maintains its integrity over time. Configuration Management (CM) ensures that the current design and build state of the system is known, good & trusted; and doesn’t rely on the tacit knowledge of the development team. It allows access to an accurate historical record of system state for project management and audit purposes. Configuration Management overcame the following challenges:

* Figuring out which components to change when requirements change.
* Redoing an implementation because the requirements have changed since the last implementation.
* Reverting to a previous version of the component if you have replaced with a new but flawed version.
* Replacing the wrong component because you couldn’t accurately determine which component needed replacing.

Let us understand its importance through a use case.

The best example I know is of the New York Stock Exchange (NYSE). A software “glitch” prevented the NYSE from trading stocks for almost 90 minutes. This led to millions of dollars of loss. A new software installation caused the problem. That software was installed on 8 of its 20 trading terminals and the system was tested out the night before. However, in the morning, it failed to operate properly on the 8 terminals. So there was a need to switch back to the old software. You might think that this was a failure of NYSE’s Configuration Management process, but in reality it was a success. As a result of a proper Configuration Management process, NYSE recovered from that situation in 90 minutes which was pretty fast. Had the problem continued longer, the consequences would have been more severe.



Now, I hope you know the importance of Configuration Management. Configuration Management stage can be considered as the backbone of DevOps. It allows more frequent software releases in the safest and most reliable way possible.

Check out this video on Puppet for better understanding.

**What is Puppet? Puppet DevOps Tutorial | DevOps Tools | Edureka**

[Become A DevOps Engineer Now](https://goo.gl/VAEh96?utm_source=blog&utm_medium=button&utm_campaign=what-is-puppet-become-a-devops-engineer-now)

Next, let us see some applications of Puppet.

**What Is Puppet – Applications Of Puppet**

Let us understand the applications of Puppet using a case study. If you are a poker enthusiast or if you have ever played online games, then you must have heard about Zynga. It is the world’s largest social game developer. Zynga’s infrastructure uses tens of thousands of servers in both public cloud and private data centers. Early on they were using a manual process, including kickstarters and post installs to get hundreds of servers online.

Now, we will see what problems they faced with this process:

* **Scalability & Consistency** – Zynga was experiencing phenomenal growth and its infrastructure needed to keep pace with the industry. Script-based solutions and manual approaches were not sufficient for their needs.
* **Portable Infrastructure** – Zynga needed a way to leverage a consistent configuration management approach in both their public cloud infrastructure and their own data centers.
* **Flexibility** – Given the diversity of the various Zynga gaming properties, it was important for the team to be able to quickly match the right configuration for the right machine.
* **Infrastructure Insights** – As the organization matured, it became more important to have an automated method of visualizing the properties of each machine.

The company was smart enough to quickly realize the need for an automated process even before they hit rapid scaling, that’s when Puppet came into the picture. Let us understand how Puppet contributes to their organization.



* **Speed of Recovery** – The production operations team can rapidly deploy the right configuration to the right box. If a system gets inappropriately reconfigured Puppet will automatically revert it back to a last stable state, or provide the details necessary to manually remediate a system rapidly.
* **Speed of Deployment** – Puppet has provided significant time savings in the way the operations team delivers services for the gaming studios.
* **Consistency of Servers** – Puppet’s model-driven framework ensures consistent deployments. According to *Mark Stockford, Vice President Production Operations*, Zynga *“It is evident that we have experienced time savings. The beauty of using Puppet is that it allows us to deliver consistent configurations across our servers in a short period every time.”*
* **Collaboration** – Having a model-driven approach makes it easy to share configurations across the organization, enabling developers and operations teams to work together to ensure new service delivery is of extremely high quality. Over a dozen people from Zynga’s team got trained in Puppet. This knowledge has been disseminated throughout the team and to the operations teams within each individual gaming studio.

After What is Puppet blog**,** [***click here to visit my next blog on Puppet Tutorial***](http://www.edureka.co/blog/puppet-tutorial/?utm_source=blog&utm_medium=content-link&utm_campaign=what-is-puppet) which will include Puppet architecture and, components of Puppet. I will also explain steps to deploy MySQL and PHP using Puppet. Puppet is the most widely used tool to achieve DevOps in organizations with large infrastructure.

[Learn DevOps From Experts](https://goo.gl/LwniqE?utm_source=blog&utm_medium=button&utm_campaign=what-is-puppet-learn-devops-from-experts)

*If you found this blog on “****What is Puppet****” relevant, check out the****[DevOps training](http://www.edureka.co/devops/?utm_source=blog&utm_medium=content-link&utm_campaign=what-is-puppet" \t "_blank)****by Edureka, a trusted online learning company with a network of more than 250,000 satisfied learners spread across the globe. The Edureka DevOps Certification Training course helps learners gain expertise in various DevOps processes and tools such as Puppet, Jenkins, Nagios and GIT for automating multiple steps in SDLC.*