LAPIN AMK
Lapland University of Applied Sciences

AUTOMATED DEPLOYMENT PROCESS WITHIN ENTERPRISE SOLUTIONS

Case Episerver

Heinänen Michael

Bachelor's Thesis
School of Business and Culture
Degree Programme in Business Information Technology
Bachelor of Business Administration

2016

LAPIN AMK
Lapland University of Applied Sciences

Abstract of Thesis

School of Business and Culture
Degree Programme in Business
Information Technology
Bachelor of Business Administration

Author Michael Heinänen Year 2016
Supervisor Vladimir Ryabov
Commissioned by Episerver AB
Title of Thesis Automated deployment process within enterprise solutions
Number of pages 52 + 11

This research focused on studying the concept of automated deployment in Web hosted applications. The work, conducted for within Episerver, had three objectives, i.e. to reduce deployment times, cost and dependency on managed services engineers; to introduce a more reliable deployment solution with the current infrastructure in order to minimize human error; and to develop an agile and secure automated deployment process for the case company.

The research presents a fully functional deployment system that was developed for the case company. The study used one of the case company's clients to illustrate the development process and as well as the end results of the new automated deployment platform. Due to sensitivity of company data, a replica deployment model was developed for use in this study. The model is identical to the system built for the case company.

This study concludes that the case company had been using an obsolete and inefficient deployment process. As a result, an automated deployment process was developed. Through using the new system, it was reported that less service requests were created, human errors were minimized, over 150 hours of manual labour were saved for the customer, and customer satisfaction increased.

Key words Automation, Deployment, Deployment Process, Enterprise Solutions, Environments and Web Applications.

CONTENTS

ABSTRACT

FIGURES

TABLES

SYMBOLS AND ABBREVIATIONS

FIGURES

TABLES

SYMBOLS AND ABBREVIATIONS

IIS	Internet Information Services
CDN	Content Delivery Network
GTM	Global Traffic Manager
UPS	Uninterruptible Power Supply
SAN	Storage Area Network
VLAN	Virtual Local Area Network
CMS	Content Management System
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IDS	Intrusion Detection System
DDoS	Distributed Denial of Service
SLA	Service Level Agreement
IP	Internet Protocol

1 INTRODUCTION

This chapter focuses on introducing the topic of this study, together with the background, motivation, and objectives of this study. Additionally, the assumptions and limitations are discussed to provide an overview of the topic.

1.1 Background and Motivation

The use of Internet and Web hosting has grown significantly in the last decade (Pingdom AB 2010). Many large enterprises have realized that it is important to leave Information Technology (henceforth IT) work to IT experts and concentrate on innovating products and services. Code deployment in web application has traditionally been performed by service engineers in hosting companies. However, with the increase of data needs by client companies, hosting companies have had to innovate too.

This thesis is an endeavour to formalize the problem experienced in daily code deployment procedures at the case company, Episerver. Episerver is a provider of digital marketing and commerce solutions, headquartered in Sweden with clients world-wide. The deployment process in place at Episerver requires an extensive amount of manual labour which can be time consuming when deploying code of the web application to several locations. The topic researched in this study involves the concept of automatization, web applications hosted in an enterprise solution, and implementation of automated processes for deploying build packages which consists of code. (Octopus Deploy 2016a; Akhil, Calton, Gueyoung, Qunyi, Wenchang & Galen 2005).

The Enterprise is a concept used when discussing very large business networks but the size can differ depending on the organization’s size due to the enhanced scalability and efficiency of systems. An enterprise web solution generally consists of two web-front servers, one admin server and one database server where one or multiple web applications are hosted.

Substantial enterprises often consist of three environments. The Quality Assurance environment is one of them where a quality analysis team makes sure the new functionality of the website work accordingly and sees whether the new

code had an impact on the existing functionality. The Staging environment, also called an integration, is a Pre-Production environment. When the code has been tested and approved on the Quality Assurance environment, the code is deployed to the Staging environment for final testing. Once testing part is done, the code is thereafter deployed to Production environment; which is the live environment. Identical environments for staging and Production are required in order to be able to test the functionality of the application thoroughly before making the final decision whether or not deploying to the Production environment. The current deployment process is a tedious and time-consuming process. From researcher's experience of working with solution architects, performance testing is required. Particularly load testing, in order to find out whether the code has any bottlenecks that could break the application when the web application is under a heavy load. Automating the deployment process would be expedited, and the code would be deployed to the various environments instantaneously. Customers' and partners would not require to wait for an engineer from Episerver support to deploy the code manually each time.

Episerver is a leading provider of digital marketing and digital commerce solutions and was also named a leader in Gartner's 2015 magic quadrant for web content management due to its fast growth over the years (Episerver AB 2015a). Due to Episerver growth over the years, the general research of the development area is to explore various deployment approaches. In addition, to find a more efficient deployment process to replace the existing manual deployment process for new and existing enterprise web solutions hosted by Episerver. Automating the entire process would be beneficial for the customers, partners and managed services engineers who perform these deployments manually. In addition, these benefits include significantly lower costs for the customers; and managed services can focus more on providing high-end customer support to customers and partners.

The personal motivation for studying the particular topic in question is due to the relevance in the field of information technology as well as the experience obtained during my position as a managed services engineer at Episerver. Furthermore, automating the deployment process takes extensive amount of load from all the managed services engineers. Implementing the automated deployment process,

the managed services engineers would be able to focus more on the high-end support that Episerver provide.

1.2 Objectives, Assumptions and Limitations

The first objective of the study is to describe the automated deployment system that was developed by the author for Episerver for the purpose of reducing deployment times, cost and dependency on managed services engineers. Achieving faster deployment times without any cost or dependency on managed services engineers would surely increase the satisfactory level of Episerver customers and partners.

The second objective is to introduce a more reliable deployment solution with the current infrastructure engaged since human errors transpire from time to time and cannot be avoided whilst performing a deployment to an enterprise solution. Each individual has made mistakes at least once in their life, everyone does make mistakes from time to time and no one is perfect.

The last objective is to form an effortless deployment process which is agile and secure. Achieving an effortless deployment process which is agile and secure is customers' and partners' ultimate goal or fantasy.

This study is presented with the assumption that the reader has professional level of knowledge of the foundation of web applications and has at least one structured definition of automatization. Since Octopus, the software used in the implementation of this work is proprietary and involves a cost for the customer. Assuming that the reader is going to evaluate the costs and benefits of the recommended automation before implementation. In addition, the customers' partners might not have knowledge of the software, which would require them to learn a new technology and that would be costly for the customer. The limitations referred to above may be an obstacle in promoting automatization to customers and partners. This development is conducted to allow customers and partners with continuous deployments to utilize the results of the study.

1.3 Structure of Work

Chapter 2 discusses the scope, questions and methodology of the study. Chapter 3 describes Episerver's uniqueness within the hosting industry. Chapter 4 focuses on the data center and presents the hardware specifications of the infrastructure where the enterprise solution are hosted in. Chapter 5 presents the design phase of the automated deployment process. Chapter 6 gives an overview of the configuration. In addition, testing and results are presented. Lastly, chapter 7 draws the conclusions by presenting and discussing the findings of the developmental work.

2 RESEARCH SCOPE, QUESTIONS AND METHODOLOGY

In this chapter the scope and questions are discussed first. Further, the methodologies used in this research are presented.

2.1 Research Scope

The scope of this research focuses on exploring the concept of automatization. Due to tedious amount of manual labour, it takes approximately up to two hours to perform a manual deployment to an enterprise solution depending on how substantial solution the customer has. In addition, back and forth communication between Episerver and customers' partners is required since the functionality of the application needs to be tested by the partner when the code has been deployed to the desired environment.

Firstly, this research focuses on exploring various deployment strategies to achieve an agile and secure solution. In addition, the deployment strategy needs to be feasible for customers hosted by Episerver to expedite the deployment process without any dependency on managed services engineers. Retaining a deployment strategy with the level of usability it is going to reduce the cost for customers and hence increase the level of satisfaction. Additionally, partners would be able to achieve faster continuous web application deployments within enterprise solutions effortlessly. Lastly, it is necessary to have a reliable solution which is fail proof since human errors transpire from time to time when performing a deployment to an environment.

2.2 Research Questions

The following three main research questions below are addressed in this study to achieve the objectives of this study.

1. Would an automated deployment process increase efficiency and productivity within a hosting company?

Through several years of observing the various manual deployment processes, the total amount of deployment requests increases with the years. Additionally,

continuous deployments need to be carried out instantaneously since customers do not seem to respect the service level agreements. This research focuses on showing the efficiency and productivity that can be achieved by automating the deployment process within substantial enterprise solutions where continuous deployments are required. In addition, the results of data were compared and given to illustrate whether the achieved deployment strategy is beneficial.

2. Why would automated deployment process which is agile and secure within substantial enterprise solutions be beneficial for Episerver customers and partner(s)?

Since Episerver customers have very high demands concerning security, these demands have to be taken into account when designing a flawless automated deployment strategy. Moreover, an agile deployment strategy would surely enable the interest to any customer hosted by Episerver. This research therefore, focuses on demonstrating the benefits when implementing an automated deployment process within substantial enterprise solutions over a one-month period. In addition, the dependency of managed services engineers would no longer be required during deployments.

3. What should customers and partners within the web application industry do practically based on the research results?

Reviewing the results helps customers and partners to determine whether there is a necessity to implement an automated deployment process for web applications within their enterprise solutions. One question that should be asked is how frequently deployments are performed for their various environments and how many deployments are performed during a month time. Comparing the results from this research gives a good overview of the benefits achievable having an automated deployment process.

2.3 Research Methodology

The researcher employed the constructive research method. Constructive research is relevant since the generic aim of this research is to improve the

current deployment process by achieving researcher’s objectives. Analyses of the empirical data were collected by observing the deployment process that was used by a client over a month period. The data obtained were critically analysed whilst developing and exploring various deployment strategies. Since the study focuses on implementing a solution to improve the deployment strategy, the constructive research method was the right approach. Another reason for using constructive research is due to the fact that the core processes guided the researcher towards the retrieval of both practical and theoretical conclusions. (Gordana 2010).

Exploratory testing in terms of techniques were used in this research, by testing various solutions to determine which solution would be feasible with the infrastructure engaged. Additionally, testing was carried out to get a thorough understanding of the systems and surrounding issues involved. Comprehensive review of the empirical work was utilized and analyzed thoroughly, which also enabled the researcher to identify the various characteristics of automatization. Observing the deployment workflow process mentored the researcher to achieve a best practice solution when deploying to enterprise solutions. Applying the techniques mentioned guided the researcher towards an improved deployment strategy. (Itkonen & Rautiainen 2005).

3 AUTOMATIZATION WITHIN WEB HOSTING INDUSTRY

3.1 Web Technologies in General

Web technologies are used daily for researching, shopping, banking and social media. Combining web technologies enables systems to communicate with each other, and offers the possibility to share resources. In order to retain such level of technology, different kinds of network infrastructures—Local Area Network, Metropolitan Area Network and Wide Area Network—need to be able to communicate with computers and networks. The communication between the sending and receiving devices should be seamless. Therefore, mechanisms should be put in place to allow the sender device to transmit a message, and allow the receiver to retrieve the message and communicate back to confirm message receipt.

According to Microsoft (Wollin 2004), some of the examples of web technologies include:

- Markup languages, HTML, CSS, XML and XSLT;
- Programming languages and technologies that are used when building web applications, JavaScript, VBScript, PHP, C#, Visual Basic .Net, Perl;
- Web Server and Server Products aid by handling requests on a network where resources are shared amongst users.
- Databases, aid to store crucial data and information on a network;
- Business applications, such as e-commerce website where security becomes a crucial part of the application.

Since the last two decades, the World Wide Web has evolved to the part where the support of the infrastructure can host countless amount of applications. Technologies within web-based systems have increased in the last decade, and new web applications are built progressively by developers. However, due to the fast growth in the last decade and constant changes in the web technologies landscape, it is difficult to take in the volume of innovative advances and to be up to date with the latest web technologies in this diverse field.

3.2 Manual and Automated Deployment within Web Application

3.2.1 Manual Deployment

Manual deployments require manual intervention, however, manually performed deployments have become a very tedious and exhausting process in hosting industry. There are numerous ways to deploy manually to web applications, any traditional web hosting provider would require the customer or partner to upload their files over to the File Transport Protocol (henceforth FTP) location (Chang 2014). The first approach for the customer is to get in contact with the hosting company to agree on a time and date when the deployment needs to be carried out.

When the code package with instructions have been uploaded to the shared FTP location, the hosting provider takes a copy of the web application code and database in case a rollback to the previous code is required. Once the backups have been taken of the code and database the code is ready to be deployed.

The steps below illustrate one of the many traditional ways of deploying code to a web application that is load balanced:

- Hosting company reaches out to the customer asking them to confirm whether the code can be deployed.
- Once customer have replied back with a confirmation, the hosting company proceeds with the deployment.
- First step, is to exclude one of the web servers within the load balanced solution, preferably the server with least amount of end users.
- Second step, waits for the end users to get directed to the other server, once the excluded server has been emptied of its users the website on the server can be stopped.
- Third step, is to remove the existing code which exists in the web root folder, thereafter the new package can be deployed to its root location.
- Forth step, start up the website and notify the customer to test the functionality of the website before continuing with the second server.

- Fifth step, once confirmation from the customer received; hosting company needs to activate the server in the Load Balancer and thereafter wait another 5 minutes for the traffic to be balanced evenly.
- Sixth step, once the traffic has been balanced evenly; the second server is going to be excluded, and now jumping back to execute step 1 to 5. Once that is done step 7 would be to inform the customer that both servers are active in the Load Balancer.

The above describes the typical manual deployment process, which is a demonstration of a minor solution which consists of 2 web-front servers and one database server. The estimated time for the deployment within a minor solution often takes 60 minutes. However, it all depends on the amount of time the customer and partner need in order to determine whether the new functionality of the website is working properly or not. Comparison of the estimated time, minor and a large enterprise solution surely gives an idea how long it is going to take to deploy towards the environment which exists of 8 web-front servers.

3.2.2 Automated Deployment

Automated deployments have increased over the last few years, and have become a hot topic amongst various web hosting companies. The drive to automate has become a huge aspect within any hosting industry and highly valued by corporates (Luping 2008, 1). Microsoft has been ahead of many large corporates within the hosting industry since the launch of its cloud based service called Microsoft Azure. One of the downsides with Azure web applications are when performing a deployment, the web application results in a brief site outage which is not acceptable within the hosting industry. However, regards the virtualized machines. There is a possibility to load balance the website but also approaches with its pros and cons.

There are a vast number of various automated deployment systems to choose from. However, from researcher’s experience Octopus is the best available .net web application system on the market. The features that comes with Octopus are astonishingly great, some of the perks that comes with Octopus are; pulling code

from the source control system, compiling code, build-related tasks such as static code analysis and running unit tests (Octopus Deploy 2016b).

To illustrate the benefits of Octopus deployment once configured accordingly for best practice, steps of the deployment process are presented below:

- Customer reach out to the hosting company to promote the package to the desired environment.
- Hosting company service desk login to Octopus and press on the promote button; Octopus deploys automatically to the application on the various servers where the website is hosted.

As illustrated above, the process is very quick and it is one mouse click away. No further intervention is required, since the Quality Assurance (hereinafter QA) and Pre-Production environment is for testing purposes. The customers and partners have access rights to deploy to both QA and Pre-Production environment.

Comparing both manual and automated deployment, the vast majority would most likely choose the automated over manual deployment due to its benefits. The benefits attainable by automating the deployment process are astonishing; frequent daily deployments can be done within minutes and no further intervention is required by the hosting company.

3.3 Automation Concept

Automating web applications should be considered if a deployment is carried out more than once daily or if there is a necessity to perform daily deployments. (Ticknor, Corcoran, Csepregi-Horvath, Goering, Hernandez, Limodin & Pinto 2001, 258). Most automated deployment systems involve installing the application itself, creating the websites' application pool and an instance in Internet Information Systems (henceforth IIS). IIS can be defined as Visual Basic application that exists within a web server and responds to requests, such as a web browser requests (Microsoft 2016). Retaining a successful consistent automation process which is error-free and secure provides hosting companies vast range of opportunities to improve customer satisfactory level.

Some of the following benefits can be retained by hosting companies' customers:

- Lower costs, no administration fees if customer or partner deploy themselves to the desired environment.
- Significant increase in response times;
- No downtime on the web applications;
- Highly secured and error free deployment process;
- Possibilities to optimize the configuration to suit web application needs.

A customer's right to promote code packages depends on the Service Level Agreement (hereinafter SLA). If SLA agreement is 99.9% the customer is not given access to promote the package of code to Production environment themselves. From the researcher's experience it is often preferred to only allow customers' partners to deploy to Quality Assurance and Pre-Production environment.

The basic concept of automation is to increase the efficiency of deployments as response to the growth of the World Wide Web. The requirements have grown within large corporates to automatize the deployment process to retain a faster and secure deployment approach.

3.4 Enterprise Web Hosting

In the early 1990s, the hosting services came to be more popular, and hosting a personal website have decreased significantly. Therefore, companies do not necessarily have to invest in the technology and hiring staff required to build an enterprise solution hosted in-house. Small, medium and large-sized corporations outsourced their web hosting department to help the corporate in question to focus their energy on the primary business. (Brudenall 2005, 262).

From the researcher's personal experience, the list below is presented to show what should be considered when choosing the best suitable hosting company:

- Budget, the price is often the deal breaker for corporates; should not be the deciding factor when choosing the best suitable hosting company.

- The focus area, what kind of specialties can be offered by turning over the hosting of the servers and web applications to a hosting company.
- Technicalities, specifications and limitations. A lot of thought needs to be put in when deciding which hosting company to choose; large solutions such as an e-commerce and rich content websites should not go with the cheapest hosting company.
- Level of technical support provided, in case of an emergency; who to call? Is anyone available if a website results in an outage?
- Various add-ons and features provided by the hosting company, what is unique within the hosting company?
- Hardware, is there enough resources on the web servers to be able to handle the web application?
- Deployment, security and service level agreements. What options are there regards deployment, level of security provided; and what kind of service level agreement can be offered?
- Scalability, is there room to grow for the business? Scalability is an important factor and should be asked when choosing the hosting company.
- Another important factor would be to review customer reviews, satisfaction and reputation. How well the service desk engineers perform when resolving various issues within customers' website(s).

The above list gives an overview of the required aspects that needs to be considered when approaching a hosting company. By taking the various aspects into account, the best suitable company is going to be chosen.

4 EPISERVER'S UNIQUENESS IN HOSTING INDUSTRY

This chapter focuses on showing the benefits of hosting websites at Episerver and the differences of other managed service providers. Further, along this chapter an overview of Episerver managed services is going to be discussed.

Episerver managed services is an important part of the success of the company over the past years. Moreover, it is also considered as the global leader of managing Episerver websites that run thousands of customers' websites worldwide. Most of the partners develop Episerver website, and use Episerver managed services to offset their Information Technology resource to save both time and money for both their end clients and business. The managed service department is geared specifically for Episerver websites, and has a unique blend of expert knowledge underpinned by a good breed and fully managed service desk.

Episerver is a gold partner with Microsoft which demonstrates that Episerver is in the top level of solution partner for Microsoft based platforms (Episerver AB 2015b). Exceptional service desk support, is one of the markets most efficient when it comes to solving customers' requests. A steady 96% of all the tickets created by Episerver's customers and partners are solved by the service desk before they have to reach 2nd and 3rd line engineers. Furthermore, customers' services such as web applications and the entire infrastructure is monitored 24/7/365 by the dedicated service desk to ensure high availability, performance and response. (Episerver AB 2015c).

The main advantage and difference between Episerver and other managed service or hosting providers is that Episerver guarantees uptime on website level. Figure 1 below gives an overview of the degree of service provided to customers. At the bottom of the figure co-location providers are found. These provide all of the features one would expect to provide a secure, resilient and available hosting environment. Above co-location providers one finds the managed hosting providers who ensure that the servers' operations systems and web applications such as Internet Information Service are up and running properly. Moving higher up the ladder this is where Episerver is truly unique, no one knows Episerver

software as well as Episerver themselves. In addition, the monitoring and supporting service to ensure the efficiency, availability and stability of customers' online presence. (Episerver AB 2015c).

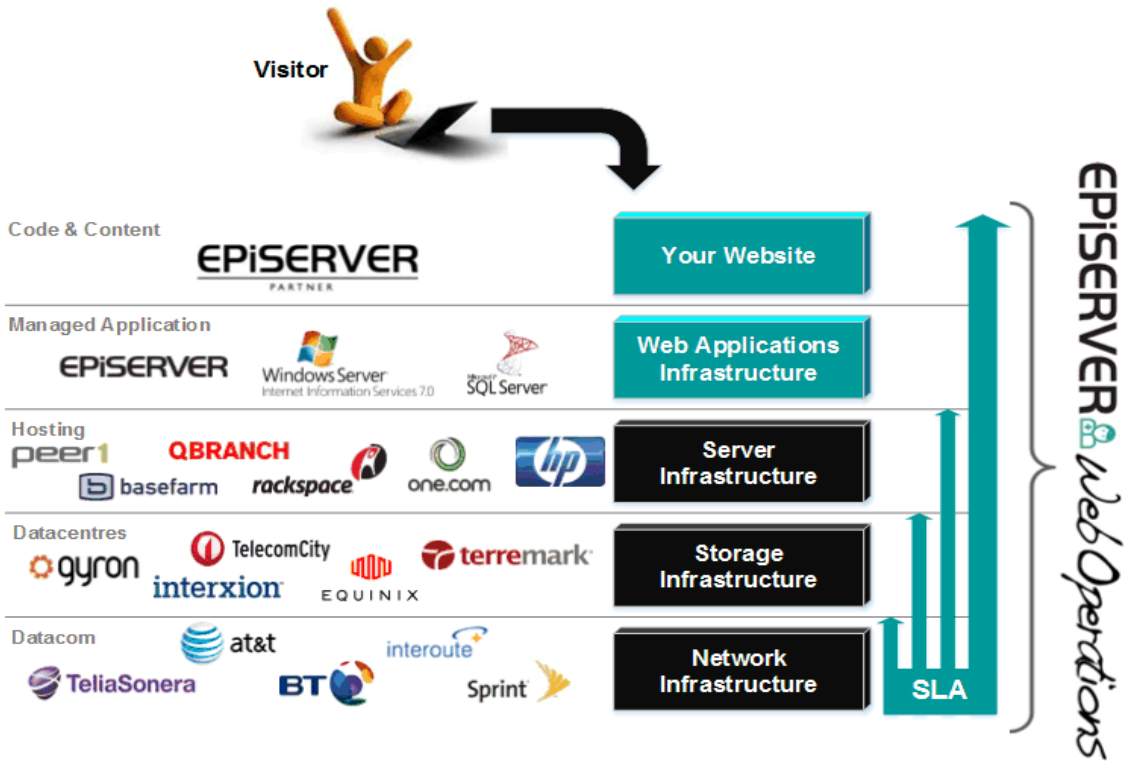


Figure 1. Types of Managed Service (Episerver AB 2015c)

The service level agreement coverage on customized code such as 3rd party applications and integrations is not a part of the SLA agreement. However, the service desk is going to assist with the troubleshooting to help isolate the area of code that may be causing the issue. (Episerver AB 2015c).

Episerver uses industries leading technology providers to deliver the same feature and level of service. Episerver receives and passes along the following level of service availability from its technology providers as shown in figure 2.

Service Function	Availability
Content Delivery Network	99.9999%
Load Balancing / Firewall	99.99%
Network	99.99%
Dedicated Instance	99.95%
Data Centre, Power & Cooling	99.99999%

Figure 2. Service Availability (Episerver AB 2015c)

As Figure 2 demonstrates, Episerver provides SLA coverage on servers, storage and network infrastructure as many other hosting providers. Moreover, what makes Episerver unique within the hosting industry is the fact that Episerver provides an SLA for their customers' websites and web application infrastructure. Figure 3 gives an overview of the SLA coverage Episerver provides for customers.

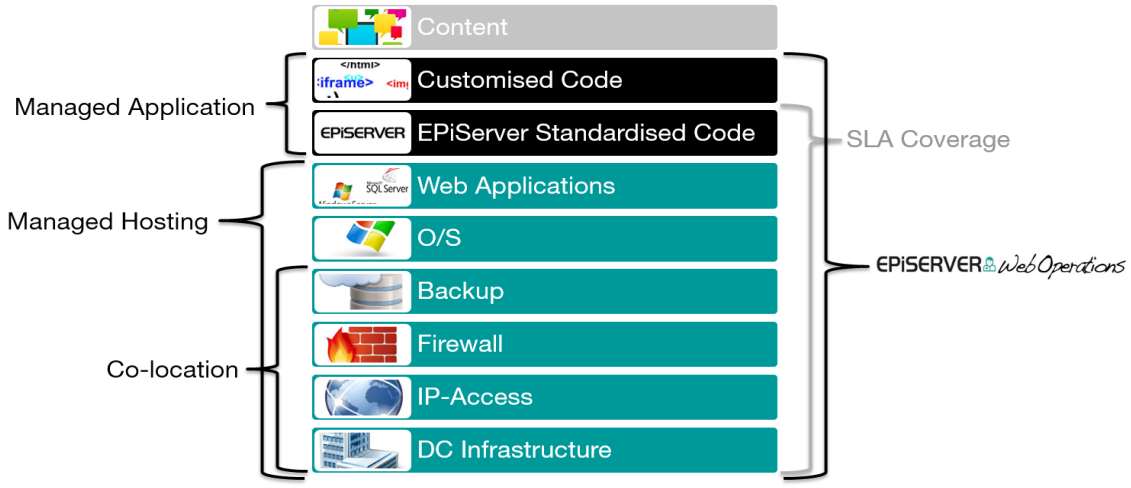


Figure 3. SLA Coverage (Episerver AB 2015c)

5 DATA CENTER, HARDWARE AND SOFTWATE SPECIFICATIONS

This chapter focuses on discussing the specifications of the infrastructure where the web applications are hosted in. Basic description is going to help retain an overview of an enterprise solution which is secure.

5.1 Data Center Specifications

Episerver has several physical separate data canters with 24 hours a day surveillance, seven days a week covering Europe, Middle East and Africa (hereinafter EMEA), Americas, and Asia-Pacific region with low latency access to data. The servers illustrated in this study have been setup in EMEA, hardware and software specifications of the servers have been presented in the appendices. Data canters which are operated use the latest technology to guarantee the highest infrastructure reliability and availability. In addition, Episerver strives to partner up with the leading green data centre providers in each market. Green data centre is a repository for data infrastructure in which the power, cooling, mechanical, lighting, electrical, and computer systems are designed for maximum energy efficiency and minimum environmental impact powered by 100% renewable energy. (Episerver AB 2015c). The construction and operation of an Episerver green data centre includes advanced technologies and strategies.

Few of the advanced technologies and strategies are listed below:

- Power provided by green energy.
- Minimizing the footprints of the buildings.
- The use of low-emission building materials, carpets and paints.
- Sustainable landscaping.
- Waste recycling.
- Installation of catalytic converters on backup generators.

The Data Centers have direct high-capacity connection to the IP network which is a part of the Internet. The hosting centers provides everything needed for a

solid and indelible operation for all the servers hosted within the Data Center. Prominent protection from power outages are provided, and the Data Centers are equipped with constant power in the form of Uninterruptible Power Supply (hereinafter UPS) and diesel generators. Hosting centers are supplied with redundant power from several different substations situated at various locations, and supplied with several different power cables via two physical paths that are routed to the hosting centers switchgear. From the switchgear electrical, the power is routed to the hosting centers power distribution units. UPS systems are based on lead acid batteries, and normally charged from the switchgear (Episerver AB 2015c).

Few fundamentals of a secure and high availability Data Center:

- IP network with redundant connections to Internet.
- Connection speeds up to 1,000 Mbps.
- Continuous power in the form of UPS and diesel generators.
- Access to data center only by key card and pin code. Each entry and exits are being logged.
- Monitoring with motion detection light and infrared sensitive video cameras.
- Temperature and humidity monitoring.
- ESD-Protected floor with cabling under the floor.
- All equipment connected to protective earth following IBM standards.

The following climate control equipment has been put in place to delimit the systems for overheating:

- Provided with district cooling which means cooled water and with municipal water as an auxiliary source.
- Controls the hosting centers temperature and humidity level.
- Temperature set to 20° Celsius and relative humidity to 50%.

Physical security, alarm response, fire and preventive maintenance fundamentals are listed below:

- Security company handle the physical security and alarm response of the data centers 24x7.
- During an alarm it is transmitted directly to the security firms command center in the event of a fire, burglary or water leakage. Climate control problems such as humidity and temperatures;
- Alarms with high priority are sent directly to police or fire department.
- Data centers proprietor's guards regularly patrol the building in which the centers are situated 24x7.
- Smoke and heat detectors are installed to detect fire;
- Periodic maintenance is performed to review and analyze the systems.
- Maintenance is performed by an external contractor and each visit is recorded and review by Episerver.

5.2 Defining Clustered Hardware

In order for to retain a superior understanding of what is discussed in next sub-chapters, therefore, the concept of cluster need to be understood.

In Information and Communications Technology the term cluster is widely used when discussing hardware such as Firewalls, Load Balancers, SQL and storage. Cluster is a set of two or more hardware configured to act as a unit. The most commonly used high availability cluster configurations are referred to as active-active and active-passive cluster (Villanueva 2015).

Active-active cluster typically consists of at least two nodes. Set of two nodes are required to make sure the same service is actively running simultaneously. The main purpose of active-active cluster is to load balance the service. Additionally, to distribute workloads across all nodes, and to hinder overload on one of the nodes.

Active-passive solution is often used when overloading the nodes is not an issue. As the active-passive implies the nodes are not active at the same time. One of the nodes are active whilst the second node is configured as passive or in standby mode.

5.3 Network Specifications

5.3.1 Clustered Firewall Specifications

The enterprise solution which was setup consists of a total four Firewalls called Clavister W50 Pro. The Firewalls discussed were used to achieve a High Availability (henceforth HA) solution. Set of two Firewalls have been setup in Data Center A, and the other two Firewalls have been setup in Data Center B. The Firewalls in both Data Centers have been configured as active-passive cluster in order to retain HA on the Firewalls. The performance and capacity specifications of the Firewall discussed in this study is presented in figure 4.

Performance* and Capacity	Clavister W50	Clavister W50 Pro
Firewall Performance (plaintext throughput)	25 Gbps	55 Gbps
IPsec VPN Performance (large packets)	5 Gbps	8 Gbps
Maximum Concurrent Connections	5,000,000	8,000,000
Maximum Concurrent IPsec VPN Tunnels	5,000	10,000
Maximum Concurrent L2TP/PPTP/SSL VPN Tunnels	5,000	10,000
Maximum Number of Users	Unrestricted	Unrestricted
Maximum Number of Routing Tables (Virtual Routers)	250	500

Figure 4. Clavister W50 Pro Performance and Capacity (Clavister AB 2015)

The connectivity specifications are listed in figure 5.

Connectivity	Clavister W50	Clavister W50 Pro
Ethernet Interfaces	1 x 1GbE (RJ45)	
Expansion Slot	Four (4) slot, supports: 8 x 1GbE (RJ45), 8 x 1GbE (SFP), 2 x 10GbE (SFP+) or 4 x 10GbE (SFP+)	
Interfaces for Management / High Availability (HA)	Yes, any Ethernet interface can be configured for Management/High Availability (HA)	
Configurable Internal / External / DMZ Ports	Yes	Yes
Local Console Port	Serial Console – RJ45	
Link Aggregation IEEE 802.1AX-2008 (Static/LACP)	Yes	Yes
Maximum Number of VLAN Interfaces IEEE 802.1Q	4,096	4,096
Support for High Availability (HA)**	Yes	Yes
Service-VLAN Interfaces IEEE 802.1ad (Q-in-Q)	Yes	Yes

Figure 5. Clavister W50 Pro Connectivity Specifications (Clavister AB 2015)

The features of the Firewall are not listed or explained. This development work does not focus on discussing the actual infrastructure configuration on advanced level. The configuration is briefly touched upon later in chapter 7 to give an overview of how the hardware that have been configured without going into advanced level of the technicalities.

5.3.2 Clustered Load Balancer Specifications

The main purpose of the Load Balancers used within the enterprise solution are specifically to manage the local network traffic. The Load Balancer divides the total amount of load evenly amongst the servers. Additionally, the Load Balancer functionality comes in handy when trying to achieve faster load times, and serve the website to the end users. The use of the features that came along with the hardware were utilized, and thereafter, configured as a flawless automated deployment process within the enterprise solution. In figure 6 an overview of the Viprion 2400 chassis specifications are illustrated.

Dimensions:	6.89" (17.5 cm) H x 17.64" (44.8 cm) W x 21.18" (53.8 cm) D 4U industry standard rack-mount chassis
Weight:	42.5 lbs. (19.3 kg) (3 blank line cards, 0 power supplies, 0 blades, 1 fan tray)
Power Supply:	AC power supply One to two 100-127 VAC (1200W)/200-240 VAC (1400W) auto ranging (80+ Gold Efficiency) 17A per input line (max) DC power supply (option) One to two 1400W 44 to 65 VDC 44A per input (max) *Note: Please refer to the Platform Guide: VIPRION 2400 on askf5.com for the latest specific power ratings.*
Operating Temperature:	32° to 104° F (0° to 40° C)
Relative Humidity:	5 to 85% at 104° F (40° C)
Safety Agency Approval:	EN 60950-1:2006, 2nd Edition Evaluated to all CB Countries UL 60950-1, 2nd Edition, CSA C22.2 No. 60950-1-03
Certifications/Susceptibility Standards	FCC Part 15 Class A VCCI Class A EN 300 386 V1.3.2 (2003-05) EN 55022:2006 + C1:2006 EN 61000-3-2:2000 EN 61000-3-3:1995 +A1:2000 EN 55022:2006 + C1:2006 Class A EN 61000-3-3:1995 +A1:2000+ A2:2005 EN 55024:1998 +A1: 2001 +A2:2003

Figure 6. VIPRION 2400 Chassis Specifications (F5 Networks, Inc 2016)

The main hardware BIG IP F5 VIPRION aids to retain an advanced automated deployment process. Set of two chassis of VIPRION 2400 series in each Data Center were required, and a total of four VIPRION 2250 blades were used in the

setup. Each of the chassis supports up to four VIPRION 2250 blades, therefore, the blades were divided between the two chassis and mounted into the chassis.

Figure 7, illustrates the specifications of the VIPRION 2250 blades mounted into the chassis.

Intelligent Traffic Processing:	2M L7 requests per second 1M L4 connections per second 14M L4 HTTP requests per second 48M max L4 concurrent connections 80 Gbps L7/L4 throughput (C2400) 155* Gbps L4, 80 Gbps L7 throughput (C2200) 1 Gbps included compression 40 Gbps maximum hardware compression Included SSL TPS: 10,000 TPS (2K keys) Maximum SSL TPS: 44,000 TPS (2K keys) Bulk crypto: 36 Gbps Note: Compression and SSL resources are allocated evenly across the number of vCMP guests set up.
Hardware DDoS Protection:	Hardware SYN cookies: 60M SYN cookies per second
Software Architecture:	64-bit TMOS
Virtualization (Max Number of vCMP Guests):	80 (4 B2250 blades, 20 per blade)
Processors:	Single Intel 10-core Xeon processor (total 20 hyperthreaded logical processor cores)
Memory:	64 GB
Hard Drive Capacity:	One 800 GB solid state drive
Network Interfaces:	One 10/100/1,000 Mbps Ethernet management port Four 40 Gigabit (or sixteen 10 Gigabit) fiber ports (QSFP+) (QSFP+ 40GBASE-SR4 100m transceivers sold separately) (QSFP+ optical breakout cable assemblies available to convert to 10 Gigabit ports) Note: Only optics provided by F5 are supported.
Power Consumption and Heat Output:	Note: Please refer to the *Platform Guide: VIPRION 2400* or *Platform Guide: VIPRION 2200* on askf5.com for the latest relevant blade power ratings.
Weight:	10.0 pounds (4.5 kg)

Figure 7. VIPRION 2250 Blade Specifications (F5 Networks, Inc 2016)

The blade specifications illustrated above give an overview of the power each of the blades beholds. In addition, the Load Balancers within the chassis use customized disaggregation, advanced clustered multiprocessing, high-speed bridges, and shares the processing load within the chassis.

5.3.3 Clustered Global Traffic Manager Specifications

The Global Traffic Manager (henceforth GTM) can be compared with the Local Traffic Manager (henceforth LTM), and both can be considered as Load Balancers within a network. However, there is a difference, and what differs is that the GTM discussed is at the top level of a Data Center whilst the LTM is setup behind a Firewall. The GTM can be defined as an intelligent name resolver which resolves domains names to IP addresses, and is often also described as intelligent DNS with security in mind. Once the GTM has resolved the domain it is going to tell which IP address to route the traffic towards. (Networks Baseline 2016). In this development work the researcher used BIG IP 1600 series, and the specifications are presented in figure 8.

Specifications	1600
Intelligent Traffic Processing:	L7 requests per second:100K L4 connections per second:60K Throughput: 1 Gbps
Hardware SSL:	Included: 500 TPS Maximum: 1,000 TPS (2K keys) 1 Gbps bulk encryption*
FIPS SSL:	N/A
Hardware DDoS Protection:	N/A
Hardware Compression:	N/A
Software Compression:	Included: 50 Mbps Maximum: 1 Gbps
Software Architecture:	64-bit TMOS
Processor:	Dual core CPU
Memory:	4 GB
Hard Drive:	500 GB
Gigabit Ethernet CU Ports:	4
Gigabit Fiber Ports (SFP):	2 optional LX, SX, or copper
10 Gigabit Fiber Ports (SFP+):	N/A
40 Gigabit Fiber Ports (QSFP+):	N/A
Power Supply:	One 300W included, dual power and DC options
Typical Consumption:	105W (110V input)
Input Voltage:	90-240 +/- 10% VAC auto switching
Typical Heat Output:	512 BTU/hour (110V input)
Dimensions:	1.75" (4.45 cm) H x 17" (43.18 cm) W x 21" (53.34 cm) D 1U industry standard rack-mount chassis
Weight:	20 lbs. (9.1 kg) (one power supply)
Operating Temperature:	32° to 104° F (0° to 40° C)
Operational Relative Humidity:	10 to 90% at 40° C
Safety Agency Approval:	UL 60950 (UL1950-3) CSA-C22.2 No. 60950-00 (bi-national standard with UL 60950) CB TEST CERTIFICATION TO IEC 950 EN 60950
Certifications/ Susceptibility Standards:	EN55022 1998 Class A EN55024 1998 Class A FCC Part 15B Class A VCCI Class A

Figure 8. BIG IP 1600 Series Specifications (F5 Networks, Inc 2013)

The benefits attained by the GTM is to protect the business from outages, and to improve the application performance. Additionally, the features provided by GTM

helps to secure DNS infrastructure by protecting the business from latest DDoS attacks to some extent. Further, the GTM ensure that the users are connected to the fastest responding Data Center to serve the website rapidly to the end users.

5.4 Security specifications

5.4.1 Content Delivery Network

Content Delivery Network (hereinafter CDN) was mainly used to serve the website assets, and to allow transmitting the content of the website in a more efficient manner. The CDN is optimized to allow the data to quickly travel between where it is stored, and point of presence closest to the end user that sent the request. Once the assets have reached point of presence location the system is going to cache the assets. The assets within a website is often referred to as images or content of the website. (Akamai Technologies 2016).

Caching the content of websites is going to enable some of the following benefits below:

- Scale rapidly without the need of worrying about the hosting capacity to accommodate any surges in demand.
- Access the world’s greatest edge network without the cost and hassle of building one in various geographical locations.
- Large website solutions such as enterprise solutions which consists of social media and e-commerce, customers may rest in ease knowing the entire website loads rapidly world-wide.
- Highly reliable service which globally delivers the content of a website in a timeless fashion regardless of the end users geographical location.

To mitigate network based attacks the protection was turned on in the CDN. The attacks are absorbed at the application layer, and deflect all traffic target at the network layer as SYN Floods or UDP Floods and to authenticate valid traffic. This built-in protection is only allowing HTTP and HTTPS traffic towards the environments. (Episerver AB 2015c).

5.4.2 Intrusion Detection System

The purpose of Intrusion Detection System (hereinafter IDS) is to monitor all events which occur in various environments. Performing analysis in search of possible incidents, which could be violations or looming threats of server security policies or standard security practices. High level of protection of all incoming traffic from the Internet towards the environments are protected, and analyzed in real-time by thoroughly analyzing the incoming network traffic. The analysis is conducted mainly to verify that communications are in compliance with applicable communication protocols. In case the verification has been confirmed the system is going to automatically continue its analyses of the traffic, and continuously compare the traffic flowing through against the database which contains traffic profiles. The traffic profiles in place describes the patterns of various known methods used when attacking a network. (Vijayarani & Sylviaa 2015).

Without IDS implemented, there would be difficulties to detect whether the customer has been the victim of a malicious attack. In addition, difficulties to prevent these malicious attacks from happening in case you do not have any information of the attacks. The careful structured approach of the IDS was introduced into the customers' network in a controlled manner ensuring that it is initially configured properly to reduce the amount of false positives and the risk of false negative attacks.

5.4.3 Intelligent DDoS Mitigation Service

One of the most common type of attacks is the DDoS attack. The attack commences directing large chunks of data traffic towards an Internet facing hardware such as the Firewall. Whilst encountering an DDoS attack the customers Internet connection could quickly become overloaded and interrupt the services which may result in a website outage. (Arbor Networks, Inc 2013).

Implementing the intelligent DDoS mitigation service. The traffic destined for the servers hosted within the environment is routed via the layered intelligent DDoS mitigation system. The service drops all malicious traffic generated by the DDoS attack before reaching the hosted environment, and aids to keep the websites and the services operational hosted within the environment. (PhishLabs 2013).

6 DESIGN

In this chapter, an overview of the infrastructure is touched upon. The workflow of the automated deployment process is illustrated via network diagrams to present an overview of the various environments setup.

6.1 Network Diagram of Infrastructure

6.1.1 Quality Assurance Environment

The network diagram in figure 9 presents the infrastructure of the Quality Assurance environment.

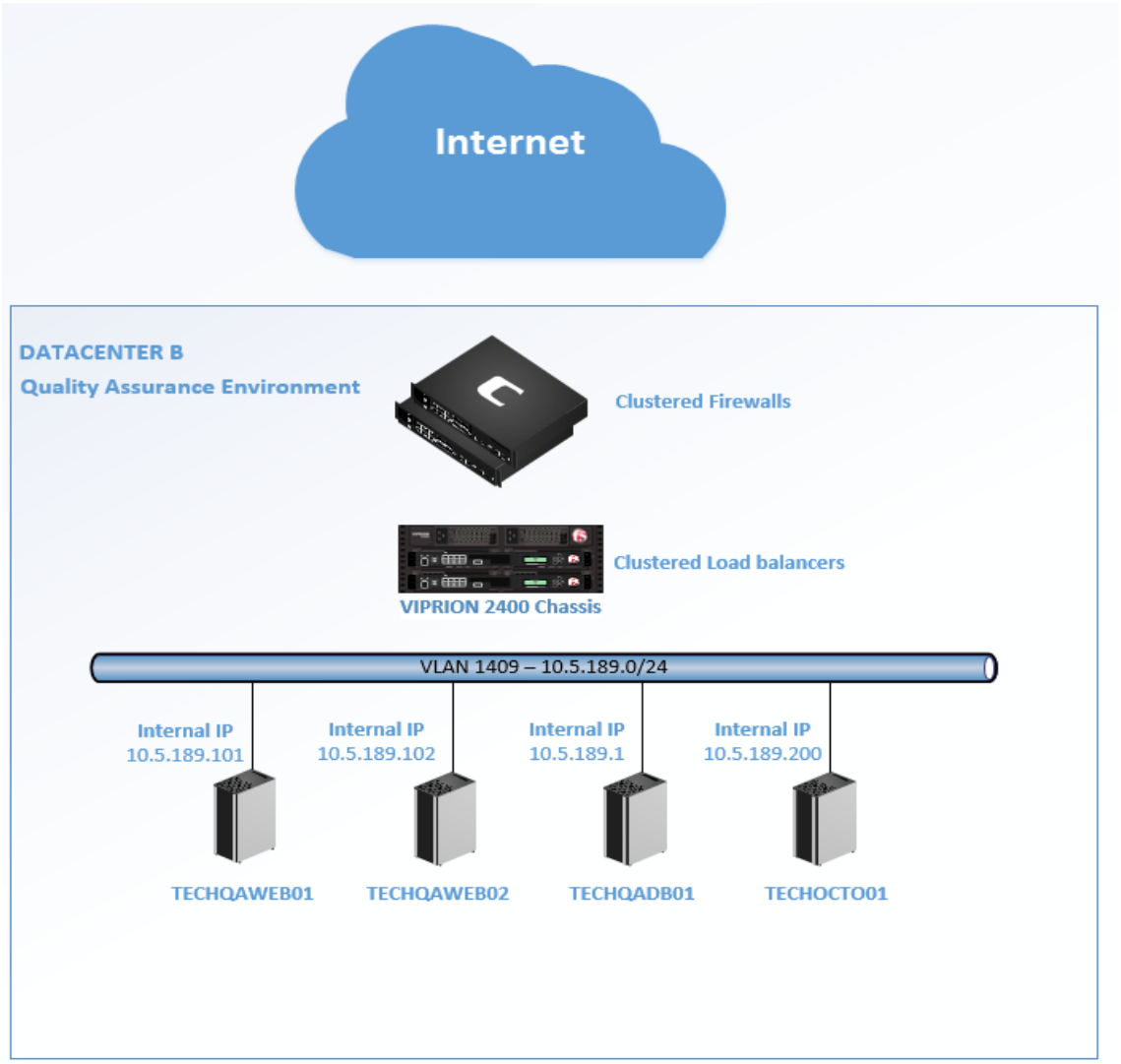


Figure 9. Network Diagram Quality Assurance

The Quality Assurance environment (hereinafter QA environment) is mainly used for code testing, the QA environment has been configured within one Data Center and not within two Data Centers as Pre-Production and Production environment. The partners developing the web applications are only accessing the web applications due to the need of code testing.

In addition, there is going to be one Octopus server setup in the QA environment. The Octopus server was linked internally to access the tentacles of the servers in both Pre-Production and Production environment. Without access to the servers, no deployments would be executed to Pre-Production and Production environment. Furthermore, in order to retain a secure solution, only one port was opened to allow internal traffic to pass through; which was required since the Octopus server needs to establish a connection with the tentacles setup on the servers within the various environments.

6.1.2 Pre-Production Environment

Figure 10 presents the infrastructure of the Pre-Production environment.

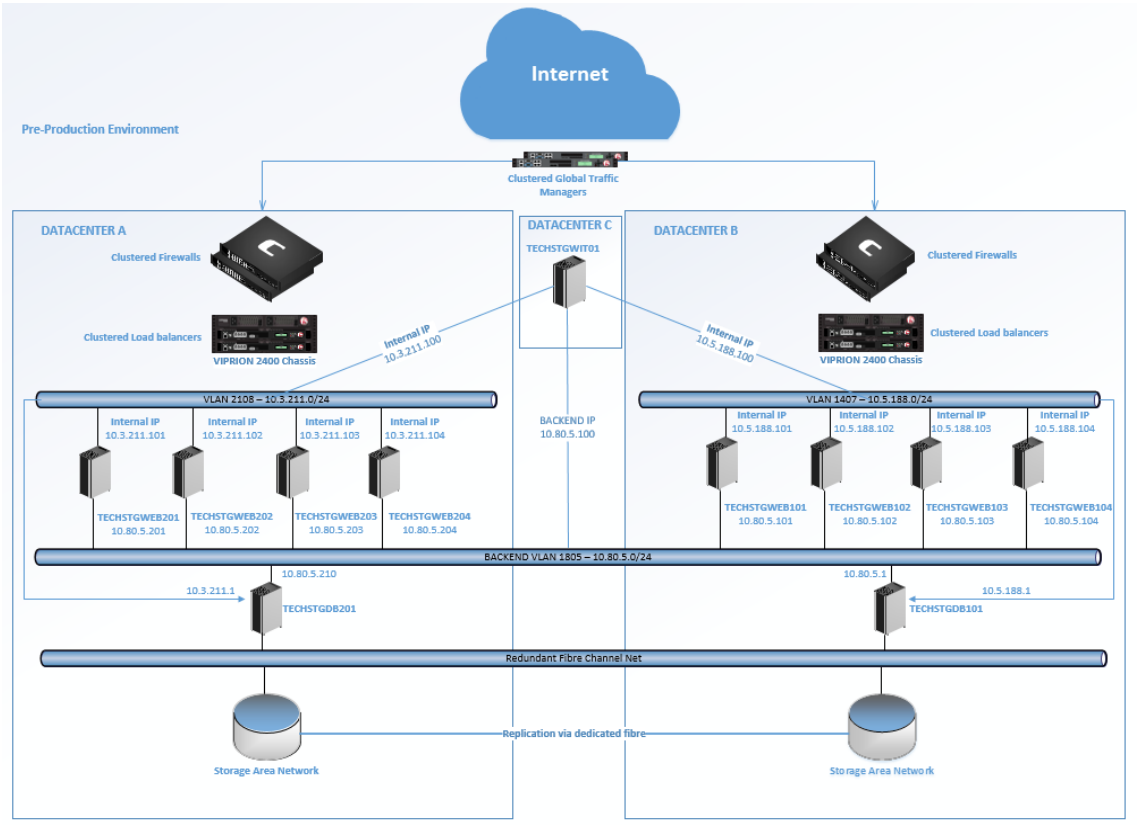


Figure 10. Network Diagram Pre-Production

High availability solution is required for partners that are constantly producing content on the websites within Pre-Production environment. Once the content changes have been approved by the customer, the content is exported and thereafter imported into the Production environment. In addition, The Pre-Production environment are also used as a final step of testing, before a decision is made whether to promote the code package to Production environment load testing is often performed to detect if there are any bottlenecks within the code.

Due to frequent content change, editing, publishing and deletion of content is done on a daily basis. The Content Delivery Network (hereinafter CDN) service was not configured within the Pre-Production environment. The downside of implementing CDN to the testing environments is that each content change would require Episerver's service desk to purge the CDN cache to make the published content visible to the client. In the case of the client with more than 300 editors world-wide whom the solution was implemented for, this would imply purging the cache every time a change is published.

6.1.3 Production Environment

The difference between the Pre-Production and Production environments is the CDN, it was only implemented to the Production environment as seen in figure 11.

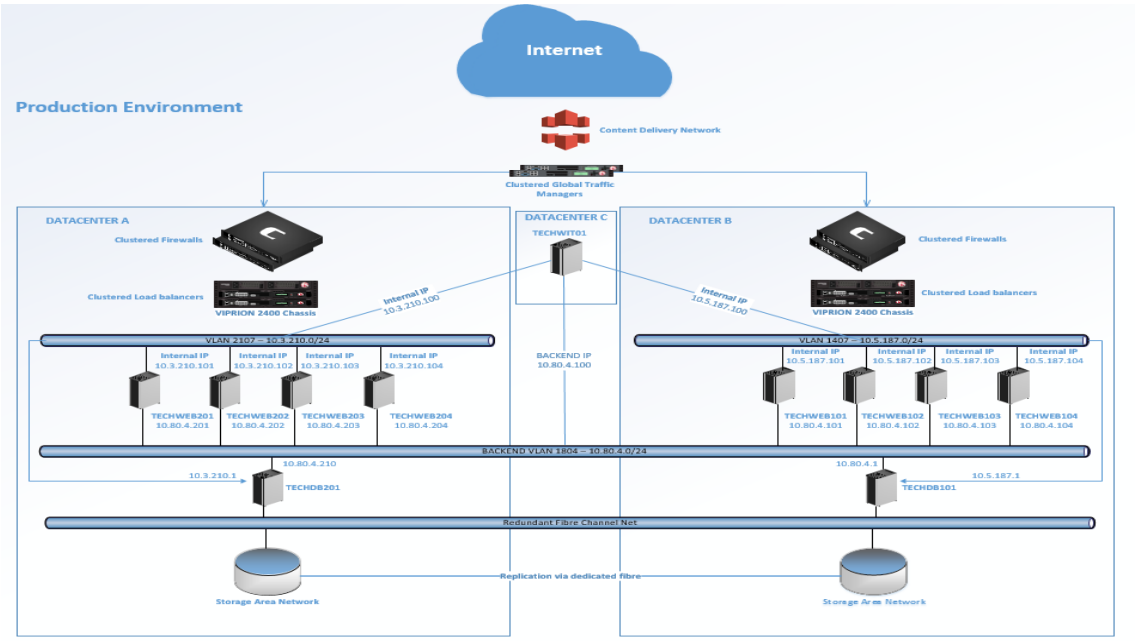


Figure 11. Network Diagram Production

The live websites are hosted within the Production environment. Security plays a big role within the environment, all of the discussed security hardware's and systems; CDN, IDS and DDoS mitigation service were implemented. It was implemented due to the cause of keeping the environment safe from unwanted incoming traffic and malicious attacks from the Internet.

There is no testing performed within the Production environment, accordingly to the clients Service Level Agreement; it is not allowed due to a guarantee of 99,9% uptime is agreed upon in the contract. Testing code within the environment could result in site outages, if bottlenecks within the code exists.

6.2 Octopus Deployment Process Workflow

The deployment process steps below is ideal for an enterprise solution when the website needs up and running without any downtime while a deployment of new code is being deployed towards the various environments.

Step 1 - Main Web Application, configured to only deploy one server at the time.

Step 1.1 – Disables the server from Load Balancers pool and waits for 2 minutes.

Step 1.2 – Stops the application pool of the website in question.

Step 1.3 - Deploys the build of code towards the website.

Step 1.4 - Copies license file from its physical stored location into the web application root folder.

Step 1.5 – Grants the application pool sufficient amount of rights to run the web application.

Step 1.6 – Starts up the web application.

Step 1.7 - Waits 120 seconds for the web application to recompile.

Step 1.8 – Tests the URL by sending a request towards the website, and expects to receive HTTP code 200.

Step 1.9 – Activates the server in Load Balancers pool.

Step 2 - Purges Content Delivery Network cache of website in Production environment.

Step 3 - Sends an email to customer and partner when deployment is done, defining the steps that were executed within the deployment.

Step one is repeated until all the servers have been deployed within the environment, once code has been deployed to all servers; Octopus thereafter continues with step 2 and lastly executes step 3. In addition, the scripts were built using PowerShell, and are presented in the appendices.

7 IMPLEMENTATION, TESTING AND RESULTS

This chapter touches briefly upon the network hardware configuration at a basic level. Further, the automated deployment process configuration is presented with the hardware engaged within the infrastructure; which was configured by the researcher.

7.1 Configuring Global Traffic Manager

The Global Traffic Manager (henceforth GTM) comes in convenient when traffic needs to be resolved two or more Data Centers. Due to clients' high expectations the GTM was required in the customers' solution. The GTM is configured to resolve domains and forwards the requests towards the any of the two Data Centers. Additionally, the CDN was configured to ensure requests of end users are directed towards the closest and best-performing environment within the Data Center.

Some of the benefits are of the GTM are presented below:

- Improved performance and availability;
- End users are directed towards the closest data center based on geographical location.
- Can manage scaling up to 40 million responses per second.
- Handles DNSSEC keys in a secure manner;
- Possibility to be configure to act as a full proxy;

The functionality of the features retained from GTM surely leaves the client and partner in ease. GTM will handle the websites requests with care and make sure its users stay connected at all times.

7.2 Configuring Content Delivery Network

Content Delivery Network (hereinafter CDN) has been configured as a global caching layer, clients' web applications assets are cached to speed up the

delivery of content to its end users. Apart from the performance and availability, CDN also offloads the traffic served directly from the origin server. Additionally, some degree of protection from DDoS attacks are absorbed of the large distributed infrastructure.

Figure 12 illustrates the functionality and shows how CDN serves the content from the edge servers.

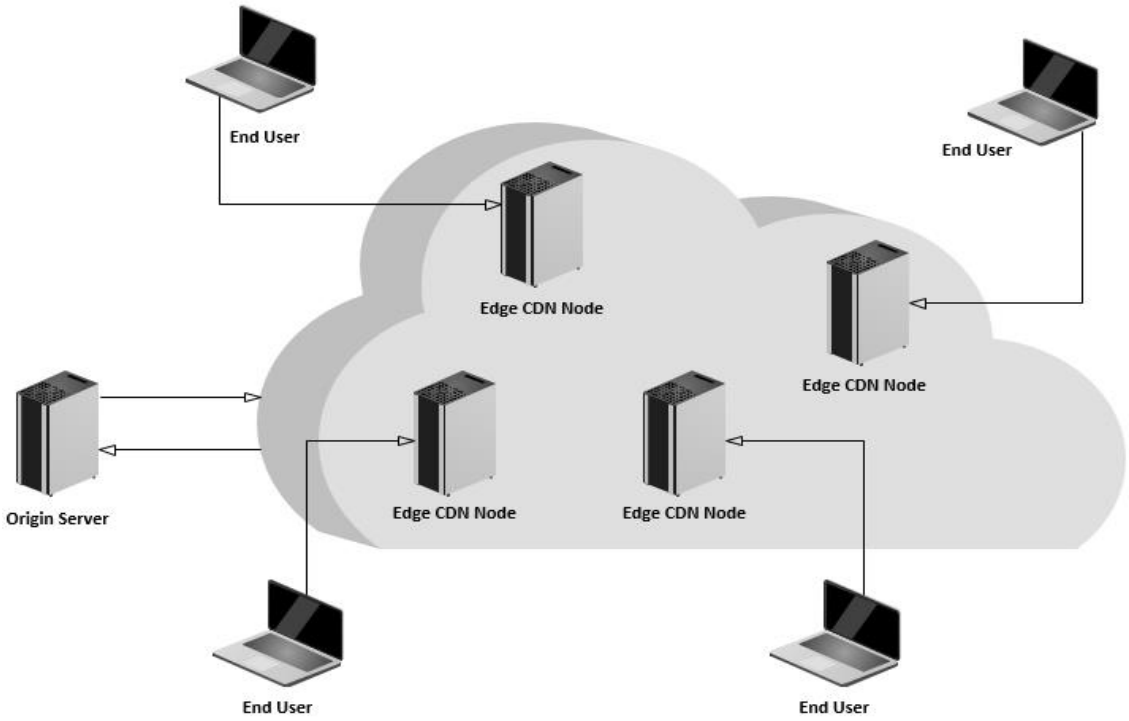


Figure 12. Content Delivery Network Functionality

The nodes also called as edges fetch content from the origin server. Upon a request the CDN caches the content, and serves the cached content to the end user from the nearest possible located edge instead of fetching the content from the origin server each time.

7.3 Configuring Firewall

The Firewall has been configured to keep out unwanted traffic and to help screen out worms, viruses and hackers from accessing the environments. Firewall do have a big key role within the infrastructure, without a Firewall present; the environments would be vulnerable. The Firewalls have been configured as an active-passive cluster since the websites hosted within the clients' solution do not

generate heavy traffic. Most of the websites hosted within this enterprise solution do have low amount of traffic which concludes the need of active-active cluster solution. Figure 13, presents the request workflow of the active-standby Firewall cluster.

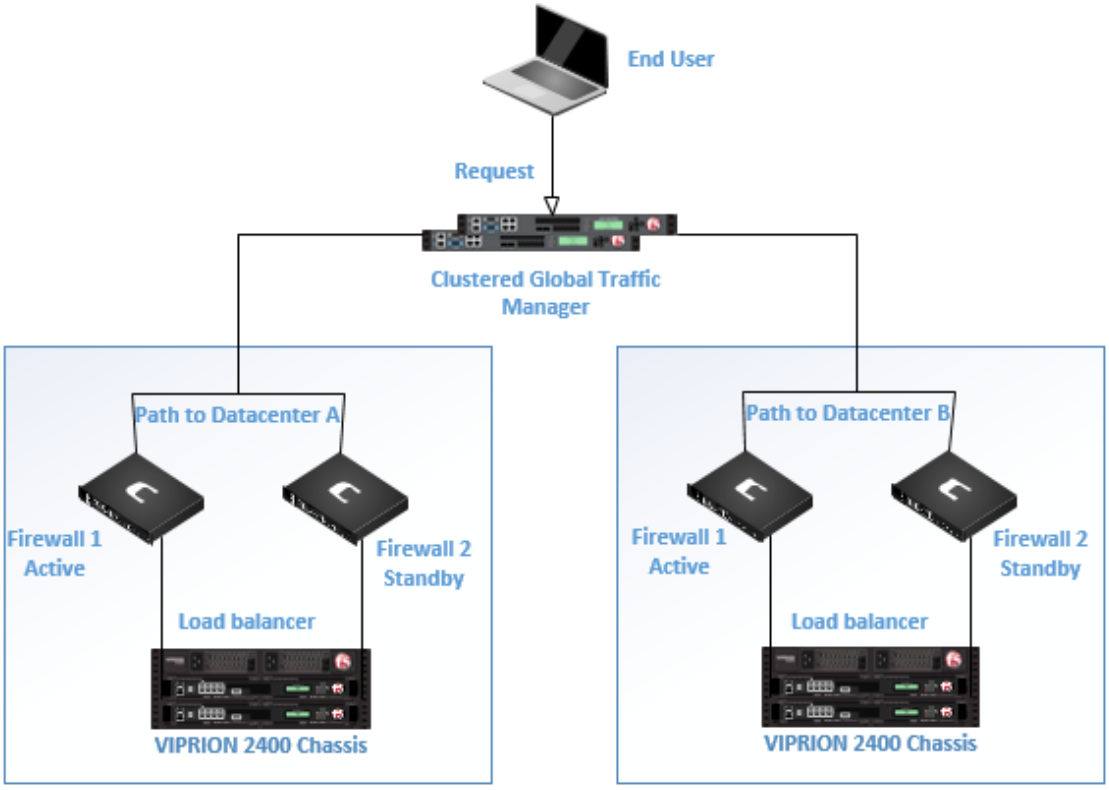


Figure 13. Firewall Configuration of the Traffic Workflow

Each of the three environments, Quality Assurance, Pre-Production and Production environment is safeguarded by the Firewall that controls both outgoing and incoming traffic. Since new threats constantly fall into the picture and new methods which do harm are introduced, the Firewall is continually monitored and administrated by both the managed services and infrastructure department. Real-time monitoring systems have been configured, to inform Episerver whether there is heavy load of incoming traffic from the Internet. There is a necessity to keep the hardware updated to ensure high level of security that complies with the modern demands on Firewalls. The various environments were configured to only allow both HTTP and HTTPs traffic and block rest of the unwelcome incoming traffic from the Internet.

7.4 Configuring Load Balancer

Same principle for the Firewall as for the Load Balancer. Due to the high availability requirement, one of the Load Balancers are set to active mode whilst the second Load Balancer is set to standby mode; and the following configuration has been applied to both Data Centers. In case of a malfunction on the current active Load Balancer the standby Load Balancer becomes active, the operations of its current services would continue functioning.

Once the initial request is forwarded to the Load Balancer it is going to be handled by the virtual server. Thereafter the Load Balancer forwards the request to the pool which consists of the servers. Since large enterprise solutions, the web applications can generate extensive amount of incoming traffic towards the servers. The pool has been configured to balance the traffic evenly on the various servers within the pool, to aid the servers from becoming overloaded with incoming traffic from the Internet.

Load Balancers have two monitors configured in each of the Data Centers. The monitors within the pool determines whether a server should be active in the pool or not. One of the features are HTTP monitor which has been created and configured accordingly. The monitor sends a request every 10th second to each server with host URL configured for the http monitor, and expects to find “viewport” in the delivered data to determine whether the site is functioning properly. In case any of the servers deliver content that do not include “viewport” the server is excluded instantaneously during an internal server error or equivalent. However, the HTTP monitor configuration varies from website to website since it is code dependant.

Include check monitor feature in the Load Balancer is the key to automating the entire deployment process. The monitor is used to exclude a server from the pool at any given time. The monitor sends a request every 10th second to each server with host URL, the include check and expects to find a htm file called “lbcheck” inside the root folder of the web application. The expected behaviour is for the monitor to find “INCLUDE=TRUE” in the delivered content, in order to keep the

node active in the pool. Any other response excludes the server from the pool in various ways as described below:

- If the value “INCLUDE=TRUE” is set in the lbcheck.htm file, the server is included in the Load Balancing pool.
- In case the value is set to “INCLUDE=DISABLE”. The Load Balancer excludes the server from the pool. However, it will not close any existing active or persistent connections immediately. Eventually, all connections are dropped; and once end users have ended up on other servers the server in question is marked as inactive. This method is time consuming but a lot safer than the method below.
- Lastly the value “INCLUDE=OFFLINE” or removal of the file is going to force the server to go offline, and exclude the server immediately resulting in dropped connections for any currently active users. The method discusses is the fastest way to exclude a server. It is not an ideal scenario as any ongoing connections are dropped instantaneously resulting in lost articles within the shopping cart of the e-commerce website. The method discussed should only be used in critical situations when you need to shut down a server immediately.

7.5 Configuring Octopus

7.5.1 Octopus Project Creation, Environments and Roles

An Octopus project can be defined as a compilation of deployment steps. Used when configuring the web applications, in which order the deployment steps will be performed. (Stovell 2015). Projects were created for each of the clients’ websites to separate from each other in order to retain a friendly user interface which would result in more user friendly administration.

Figure 14 illustrates the separation of the various environments built for the enterprise solution to be hosted in.

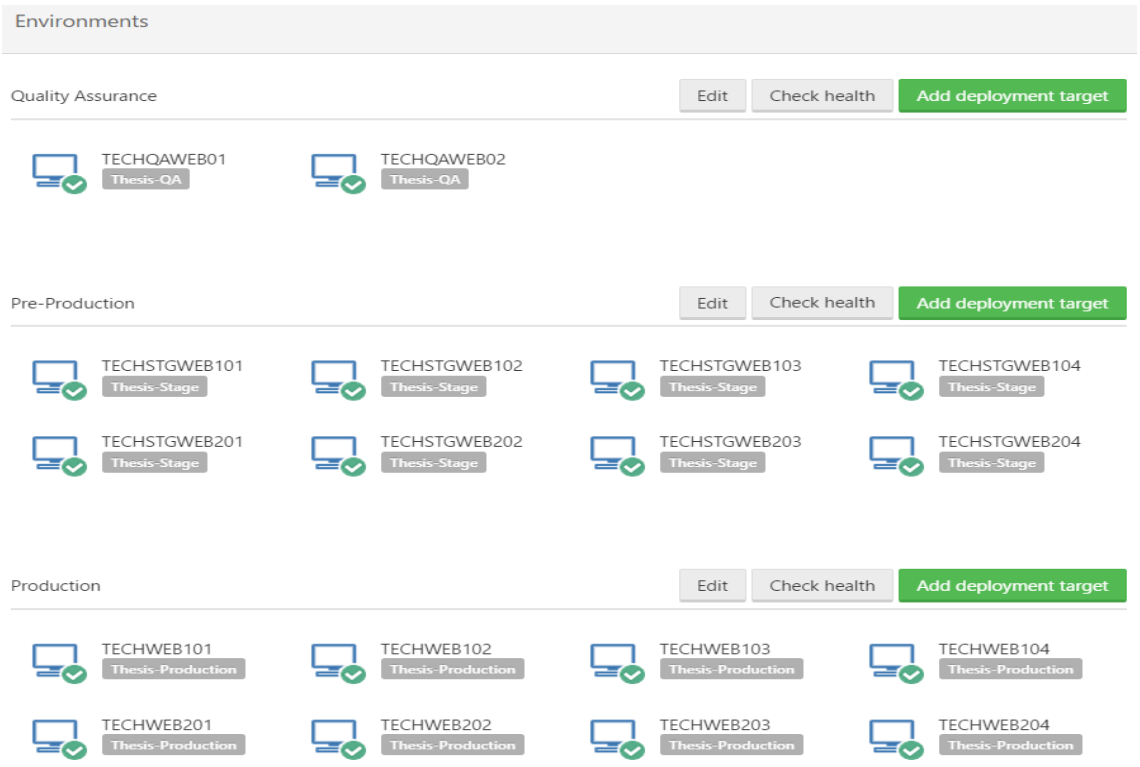


Figure 14. Octopus Overview Environments

The purpose of environments is to define which group of servers the build package of the code is going to be promoted towards. The roles presented in figure 14, Octopus functionality allows to specify which deployment step within the deployment process or set of variables to be executed towards which environment.

7.5.2 Octopus Lifecycle

The lifecycle that has been created for the thesis project is identical with the clients' solution. The projects lifecycle is divided into two phases; the first phase is going to be configured as the testing phase. Once a nuget package of the application is uploaded, Octopus automatically deploys the package to Quality Assurance environment (hereinafter QA environment) to test the code functionality before promoting the code to Pre-Production environment. However, manual intervention is required when deploying to Pre-Production environment. Users with sufficient rights to deploy, needs to promote the deployment towards the desired environment. Regardless the second phase of the lifecycle is the

Production phase. The way it is configured is to not allow any deployments to be promoted to Production environment unless the code build have been deployed to both Quality Assurance and Pre-Production environment successfully (Love 2015). Figure 15, presents the Lifecycle created for the project.

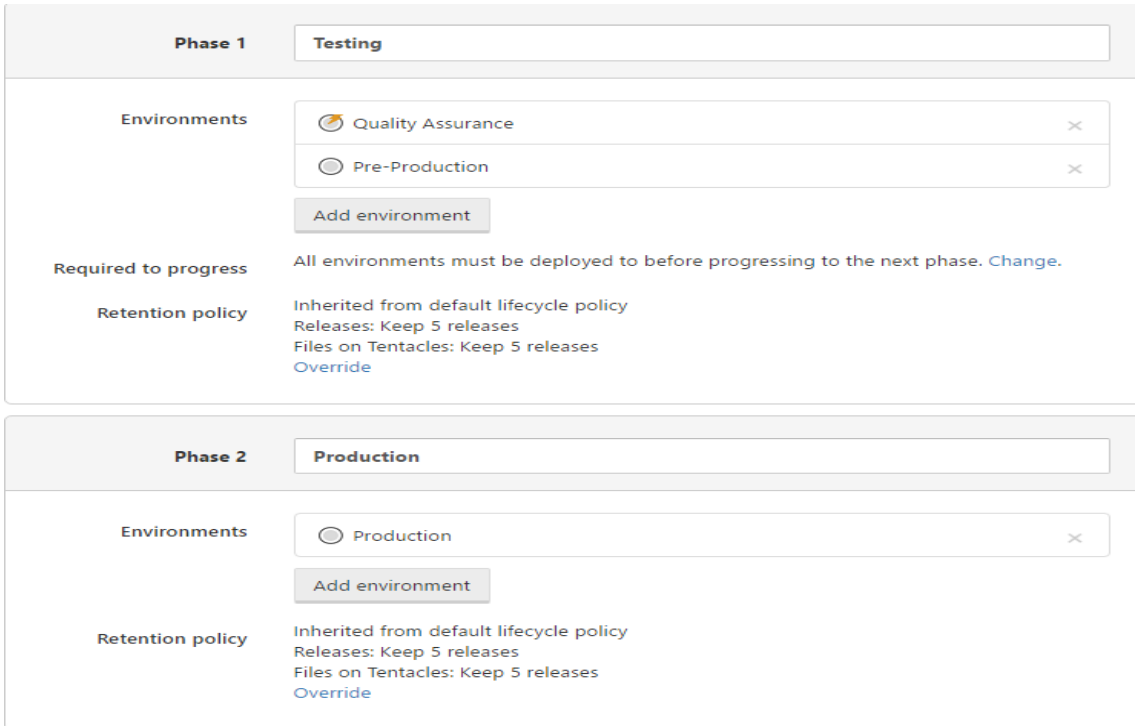


Figure 15. Octopus Lifecycle Phases

How it is configured, customers', partners' or managed services engineers would not be able to accidently promote any packages to the Production environment. The packages can only be deployed to Production environment, in case the packages have been promoted to both QA environment and Pre-Production environment successfully. The benefits with the lifecycle discussed is to make sure the code is tested thoroughly before deployed to Production environment.

7.5.3 Octopus Deployment Process

The deployment structure defined in figure 16 have been configured for the various environments to retain an automated process whilst deploying to any of the environments. The condition feature which exists within each step were configured to only execute the next step if the previous step was executed successfully.

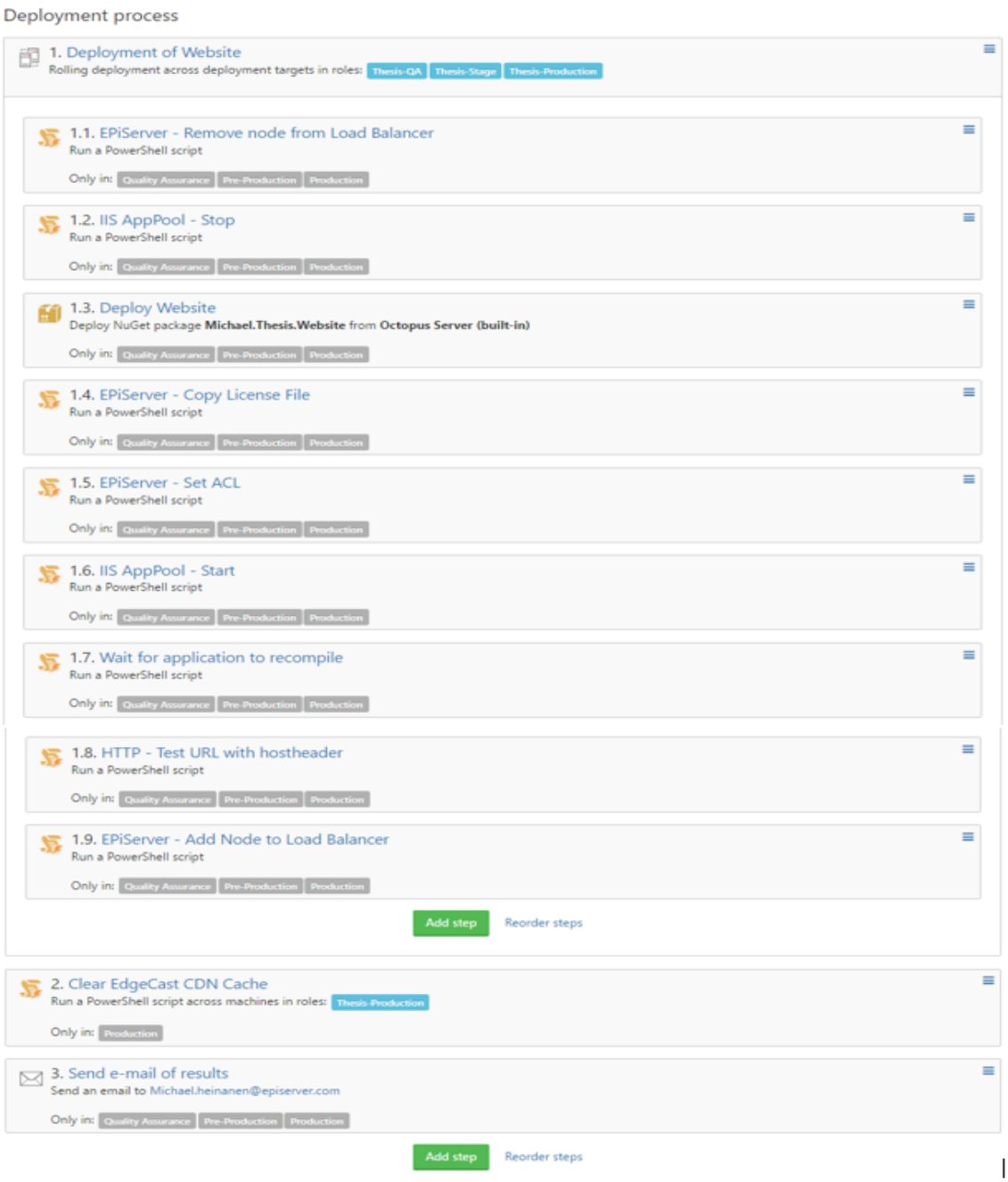


Figure 16. Octopus Deployment Process Workflow

Configuring the deployment process accordingly as presented in figure 16 enables customers, and partners to feel safe when the code is deployed. Through months of testing, each step was designed carefully to retain a user-friendly deployment approach.

7.6 Testing Phase and Results

This thesis study used a replica solution identical to one that was built for one of Episerver customers. Due to a strict company policy, the possibility to use the real company's solution decayed. Therefore, the researcher developed a solution identical to the real solution offered to the customer.

By looking at the exported data from the customer Octopus database, the results give a clear overview of how frequently used as illustrated in table 1. Promoting the package to any of the environments can be done effortlessly within minutes. Therefore, the dependency on managed services engineers to interfere when a deployment needs to be executed towards any of the existing environments.

Table 1. Total of Deployments Executed Past Month

Customer Report Past Month				
Project Name	QA	STAGEV2	PRODV2	Grand Total
Project 1	6	6	5	17
Project 2	5	5		10
Project 3	8	4		12
Project 4	6	5	3	14
Project 5	4			4
Project 6	10	5	2	17
Project 7	6	6	5	17
Project 8	3	1		4
Project 9	2			2
Project 10	5	8	6	19
Project 11	11	7	9	27
Project 12	6	1	1	8
Project 13	7	5		12
Project 14	8	2	1	11
Project 15	9	6	4	19
Project 17	6	5	2	13
Project 18	1			1
Project 19	4	3	2	9
Project 21	9	6	5	20
Project 22	4	1		5
Project 23	1			1
Project 24	2			2
Project 25	1			1
Project 26	2			2
Grand Total	126	76	45	247

As can be seen in the table 1, the number of deployments executed in the first month of automating the deployment increased significantly. The results of 43

manual deployments in the past few months show rather low amount of deployments when compared to the results of automated deployments.

To answer the question, would automated deployment process increase efficiency and productivity in general? This research showed the efficiency and productivity that can be achieved by automating the deployment process within substantial enterprise solutions where continuous deployments are required.

After adoption of the Octopus deployment process, several benefits of using an automated system were reported by both Episerver's clients and staff alike. Some of the benefits include the following:

- Less service requests created, managed services can focus more on resolving other ticket;
- Satisfied customers, especially the partners;
- No human errors;
- Over 150 hours of manual labour saved, just for one customer.

The study concludes that there was a gap between knowledge of manual and automated deployment. When Episerver proposed the solution to its clients, the reception was positive. Thus, getting the clients to adopt the solution was easy because they understood how beneficial the new system would be for them, partly because they knew that continuous daily deployments were required given the enormous number of website (up to 300) that needed to be maintained on a weekly basis.

Figure 17 depicts a graphical view of the total amount of deployments executed the past month.

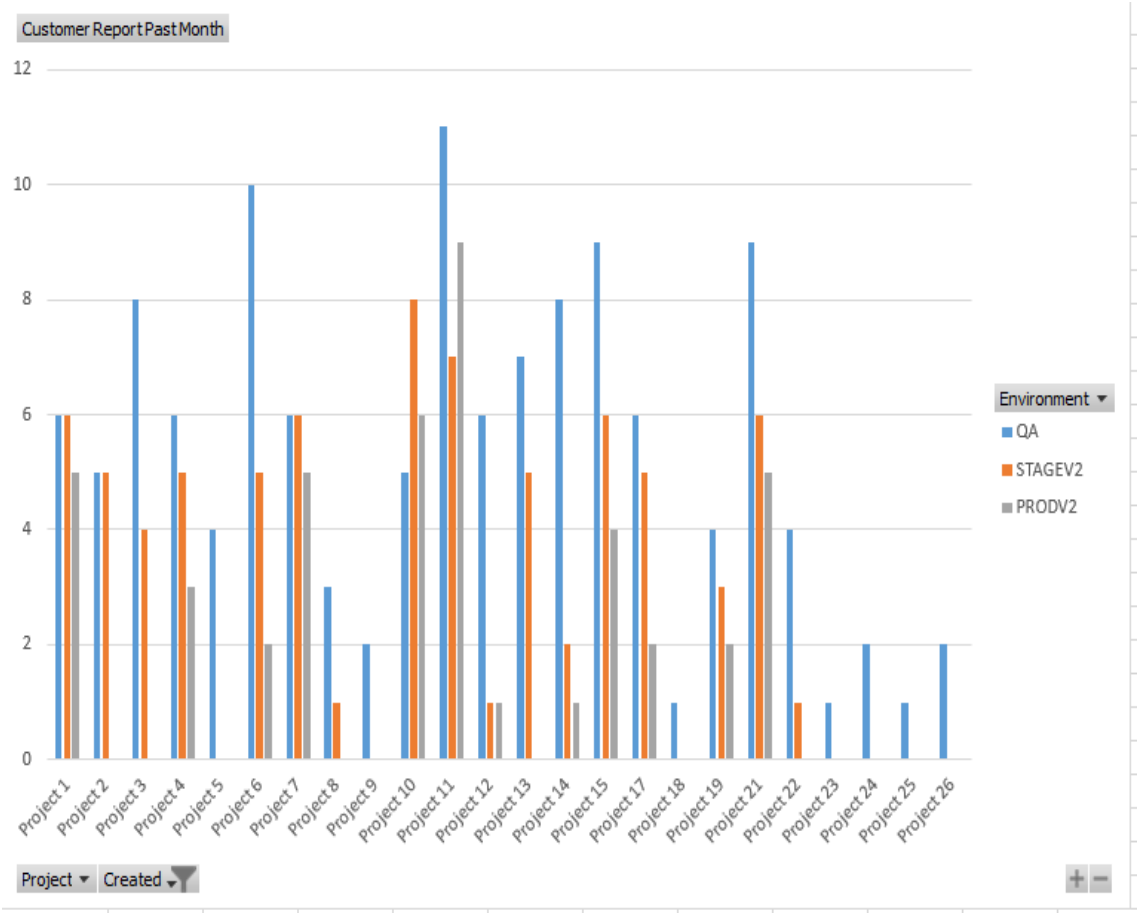


Figure 17. Octopus Deployments Executed Past Month

In consideration of the future of deployment automation, can be stated that automation is the way forward within large hosting industries. Automation allows a company to free its resources and focus them on other service requests that require human intervention. As for the future of manual deployment, such deployments are only used on some rare occasions. Generally, manually performed deployments are only recommended to customers with limited code amount of code deployments.

The automated deployment has had a tremendous impact on Episerver's processes and customer satisfaction. Improvements have been reported in services provided to customers. In addition, less tickets have been opened, which results in a reduction in manual labour.

8 CONCLUSIONS

Episerver today is known world-wide and used by various large corporates with significant data. The current manual deployment is simply not efficient given the amount of data and deployments performed at any given time. Because of this, a need arose for an automated deployment system. This research laid out the process through which an automated system was chosen and implemented in Episerver.

With the newly deployed automated system, Episerver has experienced improvements in its services. Managed services engineers can focus increasingly on providing exceptional support for customers where assistance is needed without being disturbed by manual deployments that need to be carried out if a ticket is escalated.

The introduction of the Octopus automated deployment solution and the reception received from the organization showed that even large and established organizations can sometimes be trapped in inefficient and obsolete processes. When deployments have to be carried out continuously and on a daily, manually performing this task takes a big chunk of Episerver limited resources. Having an automated system cut the deployment time by minutes.

This study concludes that any company that performs continuous deployments can benefit significantly from automated deployment processes. As with all new processes, this new deployment system was tested and no problems in terms of efficiency, security, and reliability were recorded during a one-month period of testing. Moreover, the fact that the development process of the actual automation work that took place was completely replicated in this study proves that Octopus is a relatively easy tool to use. Based on these results, it is concluded that the Octopus deployment process has significantly increased the satisfaction of Episerver customers as well as staff.

Regardless of the positive results, it is not a guarantee that every automation performed using the methods described in this thesis is going to be a success. Moreover, there remains the question of what further benefits can be achieved by automating deployment processes. Taking note that the software used in the

present automation is open-source, further research may be conducted on exploring what extra benefits may be achieved by using proprietary software. Proprietary software is often more customizable, stable and well maintained. These characteristics may or may not make proprietary software more efficient, therefore, offering more benefits or none. In addition, there exists a question of why enterprise deployment service providers are seemingly still stuck in manual deployment processes. This is another angle of research to consider.

BIBLIOGRAPHY

Akamai Technologies 2016. Powering the next generation CDN. Accessed 17 May 2016
https://www.akamai.com/us/en/cdn.jsp.

Akhil, S., Calton, P., Gueyoung, J., Qunyi, W., Wenchang & Y. Galen, S. 2005. Towards Automated Deployment of Built-to-Order Systems. Accessed 7 May 2016
http://www.cc.gatech.edu/systems/projects/Elba/pub/200510_DSOM.pdf

Arbor Networks, Inc 2013. DDoS Mitigtaion Best Practices. Accessed 7 May 2016
https://www.arbornetworks.com/images/documents/Arbor%20Insights/AI_DDoS Mitigation_EN2013.pdf

Brudenall, P. 2005. Technology and Offshore Outsourcing Strategies. Accessed 30 May 2016
https://books.google.se/books?id=UjsWDAAAQBAJ&pg=PA108&dq=Technology+and+Offshore+Outsourcing+Strategies&hl=sv&sa=X&ved=0ahUKEwi4reuKiZLNAhXmKJoKHT0uCYUQ6AEINDAA#v=onepage&q=Technology%20and%20Offshore%20Outsourcing%20Strategies&f=false

Chang, J. 2014. Finally, a guide to hosting your website. Accessed 30 May 2016
http://www.jonhmchan.com/blog/2014/4/28/finally-a-guide-to-hosting-your-website

Clavister AB 2015. Clavister W50 Product Brochure. Accessed 29 April 2016
https://www.clavister.com/globalassets/documents/resources/product-collaterals/clavister-dts-clavister-w50-en.pdf

Dodig-Crnkovic, Gordana 2010. Constructive Research and Info-Computational Knowledge Generation. Accessed 23 April 2016
http://www.mrtc.mdh.se/~gdc/work/MBR09ConstructiveResearch.pdf

Episerver AB 2015a. Episerver named a leader in gartner's 2015 magic quadrant for web content management. Accessed 22 April 2016
http://www.episerver.com/about-us/pressroom/pressreleases/episerver-named-a-leader-in-gartners-2015-magic-quadrant-for-web-content-management/

Episerver AB 2015b. Episerver, A Microsoft Gold Partner and Global ISV, To Launch Major Recrruiting Drive at WPC 2015 To Grow Partner Ecosystem. Accessed 23 April 2016
http://www.episerver.com/about-us/pressroom/pressreleases/episerver-a-microsoft-gold-partner-and-global-isv-to-launch-major-recruiting-drive-at-wpc-2015-to-grow-partner-ecosystem/

Episerver AB 2015c. Managed Service with Web Operations Customer Proposal. Unpublished internal document.

F5 Networks, Inc 2013. Deliver More Applications for More Users. Accessed 29 April 2016
http://ntype.ru/upload/riverbed/f5_big-ip-platforms-datasheet.pdf

F5 Networks, Inc 2014. BIG-IP Global Traffic Manager. Accessed 17 May 2016
https://www.f5.com/pdf/products/big-ip-global-traffic-manager-overview.pdf

F5 Networks, Inc 2016. The On-Demand Application Delivery Controller. Accessed 29 April 2016
https://www.f5.com/pdf/products/viprion-overview-ds.pdf

Itkonen, J. & Rautiainen, K. 2005. Exploratory Testing: A Multiple Case Study. Accessed 23 April 2016
https://wiki.aalto.fi/download/attachments/58922404/Itkonen_Juha_ISESE2005.pdf?api=v2

Luping, Z. 2008. An Approach to Automated Agent Deployment in Service-based Systems. Accessed 30 May 2016
https://books.google.se/books?id=gonNPeY2IC8C&pg=PA1&dq=automated+deployments+popular&hl=sv&sa=X&ved=0ahUKEwj4t9ja6ILNAhXDQpoKHY5gD8kQ6AEIRTAC#v=onepage&q=automated%20deployments%20popular&f=false

Microsoft, 2016. What is an IIS Application? Accessed 29 May
https://msdn.microsoft.com/en-us/library/aa733738(v=vs.60).aspx

Networks Baseline 2016. F5 Load Balancers: LTM Vs GTM. Accessed 17 May 2016
http://www.networksbaseline.in/2016/02/f5-load-balancers-ltm-vs-gtm.html

Octopus Deploy 2016a. The Benefits of Deployment Automation. Accessed 7 May 2016
http://download.octopusdeploy.com/files/whitepaper-automated-deployment-octopus-deploy.pdf

Octopus Deploy, 2016b. Why Octopus Deploy? Accessed 29 May
https://octopus.com/why

Phishlabs 2013. Intelligent DDoS Protection. Accessed 7 May 2016
https://www.phishlabs.com/wp-content/uploads/2013/08/Intelligent-DDoS-Protection-White-Paper.pdf

Pingdom AB 2010. The incredible growth of the Internet since 2000. Accessed 22 April 2016
http://royal.pingdom.com/2010/10/22/incredible-growth-of-the-internet-since-2000/

Stovell, P. 2015. Projects. Accessed 7 May 2016
http://docs.octopusdeploy.com/display/OD/Projects

Ticknor, M., Corcoran, A., Csepregi-Horvath, B., Goering, A., Hernandez, J.P., Limodin, J. & Pinto, S.S. 2001. IBM WebSphere Application Server V8 Concepts, Planning, and Design Guide. Accessed 30 May 2016 https://books.google.se/books?id=YKrEAgAAQBAJ&pg=PA258&dq=deployment+automation+concept&hl=sv&sa=X&ved=0ahUKEwjHwauq-_3MAhULAZoKHRSSBsEQ6AEIPTAB#v=onepage&q=deployment%20automation%20concept&f=false

Love, V. 2015. Lifecycles. Accessed 7 May 2016 http://docs.octopusdeploy.com/display/OD/Lifecycles

Vijayarani, S. & Sylviaa, M. 2015. Intrusion Detection System - A Study. International Journal of Security, Privacy and Trust Management (IJSPTM) Vol 4, No 1, 31 - 34.

Villanueva, J.V. 2015. Active-Active Vs Active-Passive High Availability Cluster. Accessed 29 April 2016 http://www.jscape.com/blog/active-active-vs-active-passive-high-availability-cluster

Wollin, L. 2004. Introduction to Web Technologies for FrontPage Users. Accessed 29 May. https://msdn.microsoft.com/en-us/library/office/aa218647(v=office.11).aspx

APPENDICES

APPENDIX 1 – Octopus Deployment Step Scripts

Octopus process step templates, each step is illustrated with script built with PowerShell or HTML. The first picture on each step shows the script and the second pictures show the parameters created for the step template.

Step 1.1 – Disables the server from Load Balancers pool and waits for 2 minutes.

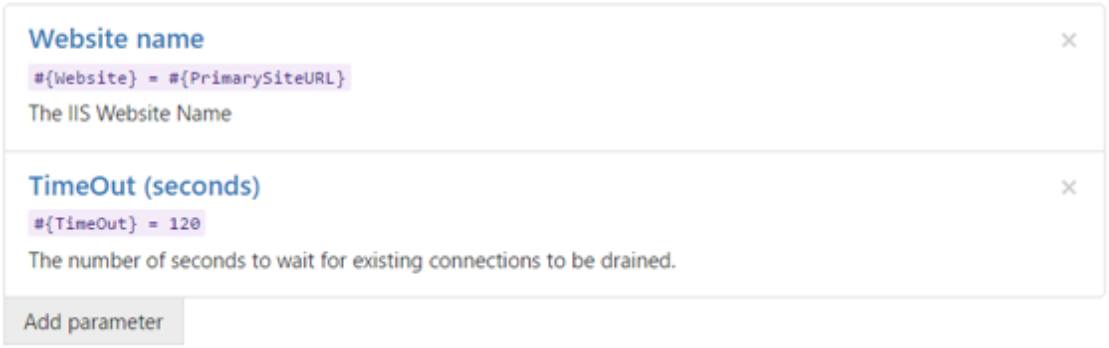
```powershell
Import-Module WebAdministration

# Get the physical path to the website
$PhysicalPath = (get-item IIS:\Sites\$website).PhysicalPath

if($PhysicalPath) {
 Write-Output "Setting INCLUDE=false in LBCHECK.HTM file in $PhysicalPath"
 "INCLUDE=false" | Out-File -FilePath "$PhysicalPath\lbcheck.htm" -Encoding ascii -Force

 Write-Output "Waiting $TimeOut for current connections to drop"
 Start-Sleep $TimeOut
} else {
    Write-Output "Could not find path to existing Web Application"
}

```



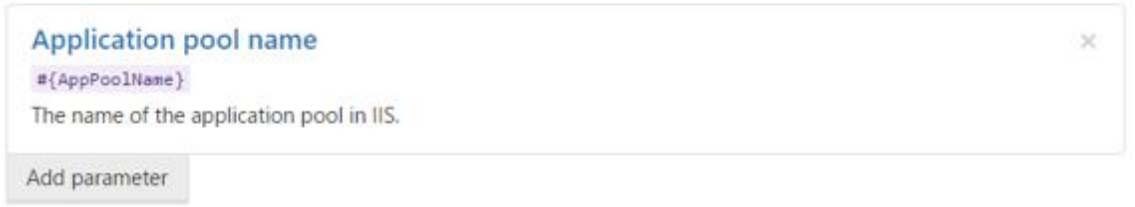
Step 1.2 – Stops the application pool of the website in question.

```
# Load IIS module:
Import-Module WebAdministration

# Get AppPool Name
$appPoolName = $OctopusParameters['appPoolName']

Write-Output "Check if $appPoolName exists"
if(Test-Path IIS:\AppPools\$appPoolName)
{

    # Stop App Pool if not already stopped
    if ((Get-WebAppPoolState($appPoolName)).Value -ne "Stopped"){
        Write-Output "Stopping IIS app pool $appPoolName"
        Stop-WebAppPool $appPoolName
    }
}
```

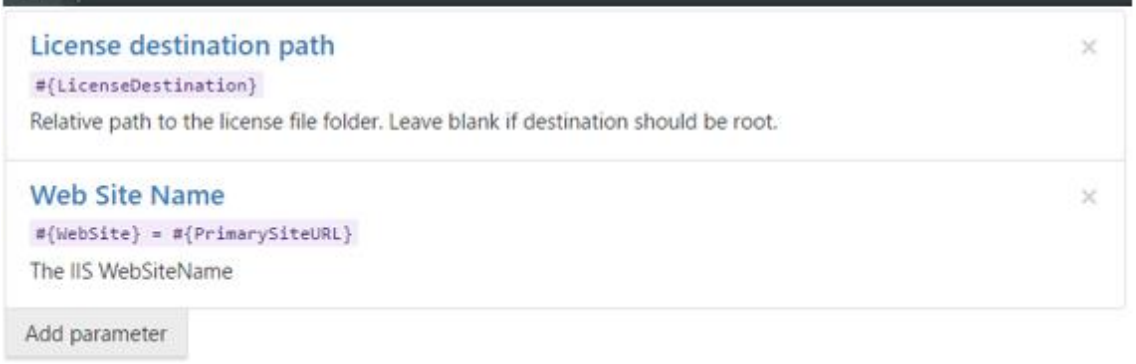


Step 1.4 - Copies license file from its physical stored location into the web application root folder.

```
Import-Module WebAdministration

$Source = "D:\Licenses\license.config"
# Get the physical path to the website
$PhysicalPath = (get-item IIS:\Sites\$website).PhysicalPath +"\"
# Get the physical destination path for license
$Destination = $PhysicalPath+$OctopusParameters['LicenseDestination']


if(Test-Path -Path $Source) {
    copy $Source $Destination
}
if($LastExitCode -gt 0) {
    exit 1
}
else {
    exit 0
}
```



Step 1.5 – Grants the application pool sufficient amount of rights to run the web application.

```
Import-Module WebAdministration
# This step template sets the ACL so that the AppPool identity created will have modify rights
# This is required by EPiServer

$AppPath = (Get-Item IIS:\Sites\$AppPool).PhysicalPath

if($AppPath)
{
 Write-Output "Setting default ACL on $AppPath for IIS AppPool\$AppPool"

 # Use the AppPath and the AppPool name to set the default ACL
 icacls $AppPath /grant "IIS APPPOOL\$($AppPool):(RX)"
 icacls $AppPath /grant "IIS APPPOOL\$($AppPool):(OI)(CI)(M)"
} else {
    Write-Output "Could not set ACL for $AppPool"
}
```

IIS AppPool name

`#{AppPool} = #{PrimarySiteURL}`

The IIS AppPool name for which the ACL should be set

Add parameter

Step 1.6 – Starts up the web application.

```
# Load IIS module:
Import-Module WebAdministration

# Get AppPool Name
$appPoolName = $OctopusParameters['appPoolName']

Write-Output "Starting IIS app pool $appPoolName"
Start-WebAppPool $appPoolName

```

Application pool name

`#{AppPoolName}`

The name of the application pool in IIS.

Add parameter

Step 1.7 - Waits 120 seconds for the web application to recompile.

```
Start-Sleep $Seconds


```

Seconds

`#{Seconds} = 120`

Number of seconds to wait

Add parameter

Step 1.8 – Tests the URL by sending a request towards the website, and expects to receive HTTP code 200.

```powershell
$uri = $OctopusParameters['Uri']
$hostHeader = $OctopusParameters['HostHeader']
$expectedCode = [int] $OctopusParameters['ExpectedCode']
$timeoutSeconds = [int] $OctopusParameters['TimeoutSeconds']

# Make verification requests
Write-Host "Starting verification requests"
$timer = [System.Diagnostics.Stopwatch]::StartNew()
$success = $false
do
{
    try
    {
        $response = Invoke-WebRequest -Uri $uri -Method Get -UseBasicParsing -Headers @{Host=$hostHeader}

        if($response.StatusCode -eq $expectedCode)
        {
            $success = $true
        }
    }
    catch
    {
        # Anything other than a 200 will throw an exception so
        # we check the exception message which may contain the
        # actual status code to verify

        if($_.Exception -like "*($expectedCode)*")
        {
            $success = $true
        }
    }
    if(!$success)
    {
        Start-Sleep -s 5
    }
}
while(!$success -and $timer.Elapsed -le (New-TimeSpan -Seconds $timeoutSeconds))
$timer.Stop()
# Verify result
if(!$success)
{
    throw "verification failed"
}
Write-Host "Verification successful"
```

URI

`#{Uri}`

The full Uri of the endpoint

Expected code

`#{ExpectedCode} = 200`

The expected HTTP status code

Timeout (Seconds)

`#{TimeoutSeconds} = 60`

The number of seconds before the step fails and times out

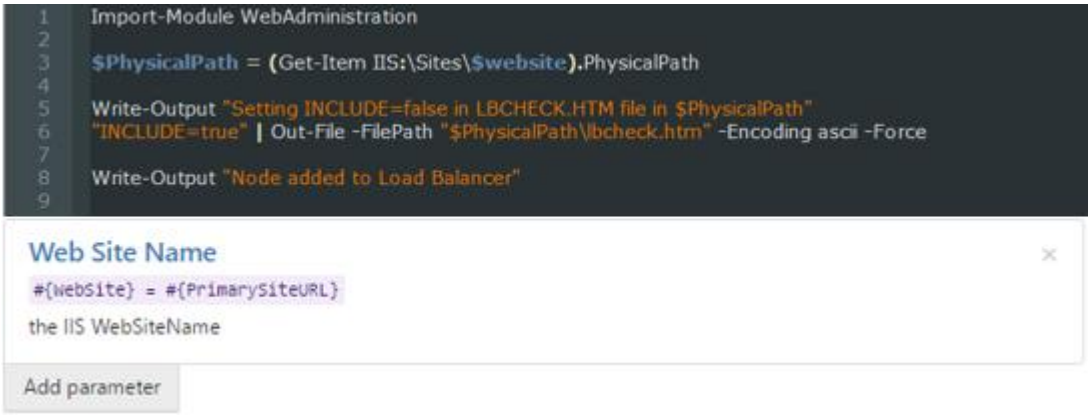
HostHeader

`#{HostHeader} = localhost`

The HTTP Request host-header to be used when making the HTTP request

Add parameter

Step 1.9 – Activates the server in Load Balancers pool.



Step 2 - Purges Content Delivery Network cache of website in Production environment.

```powershell
function Clear-EdgeCastCache
{
    [CmdletBinding()]
    Param
    (
        # CDN Account number, can be found in MCC
        [Parameter(Mandatory=$true)]
        $AccountNumber,

        # API Token
        [Parameter(Mandatory=$true)]
        [string]$ApiToken,

        # A string that indicates the CDN or edge CNAME URL for the asset or the location that will be purged from our edge servers.
        [Parameter(Mandatory=$true)]
        [string]
        $MediaPath,

        #An integer that indicates the service for which an asset will be purged. It should be replaced with the ID associated with the desired service.
        [ValidateSet(2,3,8,14)]
        [int]
        $MediaType=3
    )
    Begin
    {
        $uri = "https://api.edgecast.com/v2/mcc/customers/$AccountNumber/edge/purge"
        $headers = @{
            'Authorization' = "tok:" + $ApiToken
            'Accept' = 'Application/JSON'
            'Content-Type' = 'Application/JSON'
        }
        $RequestParameters = @{
            'MediaPath'=$MediaPath
            'MediaType'=$MediaType
        }
        $body = ConvertTo-Json $RequestParameters
    }
    Process
    {
        Write-Verbose "Request body $body"
        $request = Invoke-RestMethod -Method PUT -Uri $uri -Headers $headers -Body $body -DisableKeepAlive
        $request
    }
    End
    {
    }
}
Clear-EdgeCastCache -AccountNumber $AccountNumber -ApiToken $ApiToken -MediaPath $MediaPath -MediaType $MediaType -Verbose
```

AccountNumber

`#{AccountNumber}`

CDN Account number, can be found in MCC

ApiToken

`#{ApiToken}`

API token for accessing the EdgeCast API for the account

MediaPath

`#{MediaPath}`

A string that indicates the CDN or edge CNAME URL for the asset or the location that will be purged from our edge servers. Make sure to include the proper protocol (i.e., http:// or rtmp://).

MediaType

`#{MediaType} = 3`

An integer that indicates the service for which an asset will be purged. It should be replaced with the ID associated with the desired service., default is 3. HTTP Large

Add parameter

Step 3 - Sends an email to customer and partner when deployment is done, defining the steps that were executed within the deployment.

```
<h1>Deployment of #{Octopus.Project.Name} #{Octopus.Release.Number} to #{Octopus.Environment.Name}</h1>
<p>
 <em>Initiated by
  #{unless Octopus.Deployment.CreatedBy.DisplayName}#{Octopus.Deployment.CreatedBy.Username}#{/unless}
  #{if Octopus.Deployment.CreatedBy.DisplayName}#{Octopus.Deployment.CreatedBy.DisplayName}#{/if}
  #{if Octopus.Deployment.CreatedBy.EmailAddress} (<a href="mailto: #{Octopus.Deployment.CreatedBy.EmailAddress}">
  #{Octopus.Deployment.CreatedBy.EmailAddress}</a>)#{/if}
  at #{Octopus.Deployment.Created}</em>
</p>
#{if Octopus.Release.Notes}
<h2>Release notes</h2>
<p>#{Octopus.Release.Notes}</p>
#{/if}
<h2>Deployment process</h2>
<p>The deployment included the following actions:</p>
<ul>
 #{each action in Octopus.Action}
  <li><strong>#{action.Name}</strong> #{if action.Package.NuGetPackageId}&mdash;
  <a href="http://deploymentserver.local/packages/#{action.Package.NuGetPackageId}">
  #{action.Package.NuGetPackageId}</a> <em>version #{action.Package.NuGetPackageVersion}#{/if}</em></li>
 #{/each}
</ul>
<h3>Task summary</h3>
<ol>
#{each step in Octopus.Step}
 #{if step.Status.Code}
  <li>#{step | HtmlEscape} &mdash; <strong>#{step.Status.Code}</strong>
  #{if step.Status.Error}
   <pre>#{step.Status.Error | HtmlEscape}</pre>
   <pre>#{step.Status.ErrorDetail | HtmlEscape}</pre>
  #{/if}
  </li>
 #{/if}
#{each}
</ol>
```

APPENDIX 2 – Server Specifications

The tables below illustrate the server's hardware specifications, and software licenses required for implementing an enterprise solution which consists of three environments, i.e. quality assurance, Pre-Production and Production environment.

Quality Assurance Environment

Article	**Specification**	**No of items**
Basic System	Virtualized Server	2
Processor	vCore 2.8 GHz	4 per server
Internal Memory	34GB	1 per server
SAN Storage	100GB	1 per server

Software	**No of licenses**
Windows Server 2012 R2 Standard Edition	2
Windows SQL Server 2012 R2 Standard Edition	1
Octopus Server / Enterprise	1

Pre-Production Environment

Article	Specification	No of items
Basic System	Virtualized Server	9
Processor	vCore 2.8 GHz	4 per server
Internal Memory	34GB RAM	1 per server
SAN Storage	100GB	1 per server

Software	No of licenses
Windows Server 2012 R2 Standard Edition	9

Article	Specification	No of items
Basic System	Physical Server	2
Processor	6 core CPU	1 per server
Internal Memory	12GB RAM	1 per server
Cluster Storage	100GB	2 per server

Software	No of licenses
Windows Server 2012 R2 Standard Edition	2
Windows SQL Server 2012 R2 Standard Edition	2

Production Environment

Article	Specification	No of items
Basic System	Virtualized Server	9
Processor	vCore 2.8 GHz	4 per server
Internal Memory	34GB	1 per server
SAN Storage	100GB	1 per server

Software	No of licenses
Windows Server 2012 R2 Standard Edition	9

Article	Specification	No of items
Basic System	Physical Server	2
Processor	6 core CPU	1 per server
Internal Memory	12GB RAM	1 per server
Cluster Storage	100GB	2 per server

Software	No of licenses
Windows Server 2012 R2 Standard Edition	2
Windows SQL Server 2012 R2 Standard Edition	2