# Kubernetes

Kubernetes is an open-source platform for automating deployment, scaling, and operations of application containers across clusters of hosts, providing container-centric infrastructure. It was originally designed by Google. It has many features which are especially useful for applications running in production, like service naming and discovery, load balancing, application health checking, horizontal auto-scaling, and rolling updates.

# Kubernetes terminology

Pods – It can be one or more Docker containers that should work as a group. They should start and stop as a group. If there are multiple containers, they can reach each other via localhost. Most of the times it will be a single container within a pod.

Deployments – Deployments are objects that allow us to define the way you want to deploy your app. There are multiple strategies that we can use. These objects also contain within themselves ReplicaSets.

ReplicaSet – This is a definition of how many and what kind of pods should be running at any given time. If a pod goes down, it is a job of ReplicaSet to bring it up again.

Services – We can put a service in front of anything that needs to communicate with the outside world or within the cluster. If we create a service that can be accessed from the outside, depending on where your cluster is located, it will either assign a domain name (AWS) or an IP address (Google cloud). If a service needs to be accessed within the cluster, Kubernetes will give it an internal DNS address based on the name you give this service. It also acts as a load balancer. The way it finds the pods that it is supposed to route to is by tags.

Nodes – Nodes are the actual VM’s that is running pods.

Contents

Contents 2

Introduction 3

File Name Convention 3

Prerequisites 3

Checklist 3

Monitoring 8

Periodic Preventive Maintenance 9

Introduction

This document lists the various checks and compliances that the products should comply with before the deployment in the Production environment. The product team has to fill up the requirements and send it to the infrastructure/operations team at least 24 hours before the deployment in production.

File Name Convention  
The file should be named with the following convention  
<date in yyyymmdd\_ProductName\_TypeOfInstallation\_TargetDateOfInstallation>.docx  
  
Example: 20121231\_Lens\_Upgrade\_20130120.docx  
  
Owner   
Infrastructure Support is the Owner of this document. Changes to the primary content can be done by the Infrastructure Support team. If the product teams want any modifications then they have to send a request to the Infrastructure Team.

Prerequisites

The hardware resource requirement has to be sent by the product team to the infrastructure team at least 6 weeks in advance so that adequate resources can be provisioned.

Checklist

1. What is the task/product?

This task is to configure LENS api services with A10 Load balancer.

1. What is the schedule of installation/upgrade (Date and hour of the day and time window in EST)?
   1. Scheduled Start Time:

26th AUGUST 2016, 02:00 AM EDT

* 1. Scheduled End Time:

26th AUGUST 2016, 06:00 AM EDT

* 1. Cutoff time beyond which the deployment will be rolled back:  
     15 minutes

1. Was the client informed about the upgrade? If they were, what was the time window given to them?

No client has been informed regarding the configuration. Traffic will be redirected to the new web servers once after the Load balancer configuration.

1. What is the location (Server name: Physical path) where the programs will be deployed?

LENS api & Portal

PRODWEBSVR015 (10.0.3.17) & PRODWEBSVR016 (10.0.3.18)

LENS api & Portal - D:\websites\lensapi.burning-glass.com\

Load balancer

Prod-LB-01 (10.0.4.41)

1. Is there a Signoff from the testing team that the features were tested in demo/testing?

Yes. The QA team did the testing and all the functionalities are working as expected.

Please refer the Test report in the following confluence page.

<http://boswebsvr009:8090/confluence/display/DEV/LENS+api+Load+Balancing+Solution+-+Phase+I+Testing>

1. What are the different services that comprise the application?

<http://boswebsvr009:8090/confluence/display/DEV/LENS+api+-+Hosted+Machine+details>

1. Are automated checks in place to notify if the service fails?

Pingdom alert has been configured for LENS api parsing and search. Solarwinds configured for monitoring the Load balancer.

*(There should be two levels of checks:*

***Level 1 check and automated alerts****- This is the check that should be done from outside the LAN to check the availability of the service/site as the end user would see. Example: Pingdom alerts. Individual instance of the deployment has to have individual Level 1 checks.*

Pingdom alert has been configured for LENS api parsing and search. Solarwinds configured for monitoring the Load balancer.

***Level 2 check and automated alerts*** *- These are the checks that should be conducted on the individual services and components. If Level 1 fails then it will be because of failure of any one of the components. Level 2 checks are supposed to run on the individual components from within the LAN. Example: There should be automated checks and alert system to check the DB connectivity, Lens availability etc. Level 2 checks can be common to multiple instances of an application)*

Pingdom will check the availability of database and LENS servers periodically

1. Is the setup/patch update Readme file tested and available? Is the setup of the failover instance included in the readme file?

Load balancer configuration instructions can be found at <http://boswebsvr009:8090/confluence/display/DEV/LENS+api+-+Load+Balancer+configuration+setup>

LENS api - Readme file location <http://boswebsvr009:8090/confluence/display/DEV/LENS+api+-+Release+Notes>

The same instructions will be used to setup failover instance.

Note the following:

* 1. *With respect to the installation packages and instructions for database changes – such as database creation script, creation of users, assigning of permissions, creation of domain level users, shares and permissions to be granted, DNS names, public IP addresses to be exposed, firewall ports to be opened are to be included. So the product team should provide database changes, executable (binaries)/deployment artifact changes and system/infrastructure configuration changes.*   
       
     Does the procedure comply with the above condition?

NA

* 1. *Make sure that the permissions being used by the accounts are the minimum that are required. Avoid “Server Administrator” group permission granted to DB users. dbo privileges should not be granted to the user id being used by the application to connect to the database. “Everyone” being granted permission for file system folders/shares may not be acceptable.*  
       
     Does the procedure comply with the above condition?

NA

* 1. *Does the installation instruction include an installation verification procedure? i.e. steps to verify that the installation has completed successfully and working as expected*

Yes. Please refer http://boswebsvr009:8090/confluence/display/DEV/LENS+api+-+Installation+verification+procedures

1. What is the location of the program in the fail over server? This should include failover of individual services.

CORWEBSVR012 (10.0.2.158)

LENS api Portal - [D:\websites\lensapi.burning-glass.com\](file:///D:\websites\lensapi.burning-glass.com\)

1. What is the backup location of the programs?

Primary - [\\diskstation02\backup\CORWEBSVR011\](file:///\\diskstation02\backup\CORWEBSVR011\)

Failover - [\\diskstation02\backup\CORWEBSVR012\](file:///\\diskstation02\backup\CORWEBSVR012\)

Instructions have been given to TechOps

1. Is there a document available with the release notes for this release?

Confluence URL - <http://boswebsvr009:8090/confluence/display/DEV/LENS+api+-+Release+Notes>

1. Is there a document available with steps to failover if the primary server/s fails?

The traffic to load balancer should be redirected to CORWEBSVR011/CORWEBSVR012 in lensapi.burning-glass.com

1. What is the version of the program?

LENS api – 2.0.16.2

Load balancer - 4.0.1 build 214

1. What are the GIT details of the program? Please specify GIT URL for the tag and the location of the GIT repository.

GIT URL - <http://bossrcsvr004.usdev.burninglass.com/MM-SaaS_LensWebAPI.git>

Tagged version URL

2.0 - <http://bossrcsvr004/Repository/MM-SaaS_LensWebAPI/Tree/V2.0.16.2-Production_Release>

1. Who is the Primary owner/contact for this application?

Silambarasan Kandasamy - skandasamy@burning-glass.com

1. List of stakeholders who will get affected either by the upgrade or during the process of the upgrade.

NA

1. URL that would be affected/added

**LENS api - V1.0**

<http://lensapi.burning-glass.com/v1.0/adminservice>

<http://lensapi.burning-glass.com/v1.0/parserservice>

<http://lensapi.burning-glass.com/v1.0/adminservice.wsdl>

<http://lensapi.burning-glass.com/v1.0/parserservice.wsdl>

**LENS api - V1.7**

<http://lensapi.burning-glass.com/v1.7/adminservice>

<http://lensapi.burning-glass.com/v1.7/parserservice>

<http://lensapi.burning-glass.com/v1.7/adminservice.wsdl>

<http://lensapi.burning-glass.com/v1.7/parserservice.wsdl>

**LENS api - V2.0**

<http://lensapi.burning-glass.com/v2.0/adminservice>

<http://lensapi.burning-glass.com/v2.0/parserservice>

<http://lensapi.burning-glass.com/v2.0/searchservice>

<http://lensapi.burning-glass.com/v2.0/adminservice.wsdl>

<http://lensapi.burning-glass.com/v2.0/parserservice.wsdl>

<http://lensapi.burning-glass.com/v2.0/searchservice.wsdl>

**LENS api Portal**

[http://lensapi.burning-glass.com/portal/](http://lensapi.burning-glass.com/portal/login)

**Load Balancer**

<https://10.0.4.41>

1. Attach the load test report with a summary of maximum load the application is capable of handling.

NA

1. What is the Email server used if any (Confirm that application does not use Rackspace as the email server)?

internalrtmail

1. Confirm that the burning-glass.com domain name is not used in the sender’s email address of the application

Yes. We are not using burning-glass.com domain name.

Monitoring

1. Name the document that lists the steps for daily monitoring of the application. Reports have to be generated based on the various monitoring activities. The monitoring should involve but not restricted to the following
   1. Application logs and errors
   2. Event logs and errors specific to the application
   3. Daily usage
   4. Average response time
   5. Periodic hardware resource usage

*The monitoring activities have to be performed by the product team till the time all unknown stability issues are addressed and the application is certified as stable.*

This update does not impact any of the previously established maintenance procedures.

Periodic Preventive Maintenance

1. Name the document that lists the steps for periodic preventive maintenance of the application. The maintenance activities should involve but not restricted to the following
   1. Backup and archival of application log files
   2. Backup and archival of the DB
   3. Renaming and creation of DB tables if any

*The periodic maintenance activities will be performed as specified by the product teams by the Infrastructure/Support team by following the steps in the document.*

This update does not impact any of the previously established maintenance procedures.