

Installation

LENA Support

Version 1.3.3.0

Table of Contents

| | |
|--|----|
| 1. OVERVIEW..... | 1 |
| 1.1. mn - 1 | 1 |
| 1.1.1. Server | 1 |
| 1.1.2. Agent, Advertiser | 1 |
| 1.1.3. Manager | 1 |
| 1.2. Mechanism | 2 |
| 1.3. E F Asset..... | 4 |
| 1.4. ¾ ¤ - m` a | 5 |
| 2. Architecture / O` a | 6 |
| 2.1. Õ¹ | 6 |
| 2.2. Container ()] ^ 5 6. N_` a..... | 7 |
| 2.2.1. Container ¥ÕÖ B ()] ^ Pn | 7 |
| Kubernetes | 7 |
| ECS..... | 12 |
| Container ¥ÕÖ B Pn pl | 13 |
| 2.2.2. LENA Server bc B () · ¸ | 13 |
| 2.3. F k N_` a | 14 |
| 2.3.1. OS..... | 14 |
| 2.3.2. JDK | 14 |
| 2.3.3. ^ € User | 14 |
| 2.3.4. Library | 14 |
| 2.4. Server bc B N_` a Ð Manager..... | 15 |
| 2.4.1. × U..... | 15 |
| 2.4.2. ` æ..... | 16 |
| Memory..... | 16 |
| Disk | 16 |
| 2.4.3. 3O..... | 17 |
| " ñ#è..... | 17 |
|] ^ ĩ • | 17 |
| Directory mΦ..... | 18 |
| Log & Dump 5j | 18 |
| Health Check | 19 |
| 2.5. Server bc B N_` a Ð Session Server..... | 19 |
| 2.5.1. × U..... | 20 |
| 2.5.2. ` æ..... | 20 |
| Memory..... | 20 |
| Disk | 20 |
| 3O | 20 |
| " ñ#è..... | 20 |
|] ^ ĩ • | 21 |

| | |
|--|----|
| Directory mΦ | 22 |
| Log | 22 |
| Health Check | 23 |
| 2.6. Server b c B N_` a Ð WAS | 23 |
| 2.6.1. x U | 23 |
| 2.6.2. ` æ | 23 |
| Memory | 23 |
| Disk | 24 |
| 2.6.3. 3 O | 24 |
| " ñ#è | 24 |
|] ^İ • | 24 |
| Directory mΦ | 26 |
| Log & Dump 5j | 27 |
| Health Check | 28 |
| Server Configuration h i | 28 |
| Container Image Build | 29 |
| Application x U | 30 |
| 2.7. Server b c B N_` a Ð Embedded WAS | 30 |
| 2.7.1. x U | 30 |
| 2.7.2. ` æ | 31 |
| Memory | 31 |
| Disk | 31 |
| 2.7.3. 3 O | 31 |
| " ñ#è | 31 |
|] ^İ • | 31 |
| Directory mΦ | 33 |
| Log & Dump 5j | 33 |
| Health Check | 34 |
| Container Image Build | 35 |
| Application x U | 35 |
| 2.8. Server b c B N_` a Ð Web Server | 36 |
| 2.8.1. x U | 36 |
| 2.8.2. ` æ | 36 |
| Memory | 36 |
| Disk | 36 |
| 2.8.3. 3 O | 36 |
| " ñ#è | 36 |
|] ^İ • | 36 |
| Directory mΦ | 38 |
| Log | 39 |
| Health Check | 39 |
| Server Configuration h i | 39 |

| | |
|--|----|
| Container Image Build. | 39 |
| 3. 3.4 Fk` a | 40 |
| 3.1. 3.4 P{ | 40 |
| 3.2. Base Image ° n | 40 |
| 3.2.1. Base Image ° n Õ¹ | 41 |
| Dockerfile ‡ n | 41 |
| Dockerfile ‡ n (Û&©O) | 44 |
| Docker Image OS | 44 |
| Docker Image —ž | 45 |
| 4. Kubernetes %& × U. | 46 |
| 4.1. × U Fk` a | 46 |
| 4.1.1. × U P{ | 46 |
| 4.1.2. × U ^ € | 46 |
| ‡ ´ namespace: 3O. | 46 |
| Kubernetes Resource × U > ´ " † ñ | 46 |
| Kubernetes Resource: ŸE | 47 |
| Workload ´ " † ñ > • e | 47 |
| × Uo Resource: de | 47 |
| 4.2. Manager × U. | 48 |
| 4.2.1. 3O a | 48 |
| %! 3O a | 48 |
| \@¾° /O a Ð Workload h (.... | 48 |
| \@¾° /O a Ð Serviceh (.... | 49 |
| 4.2.2. Manifest %& × U. | 49 |
| Workload. | 49 |
| Service. | 51 |
| 4.2.3. Manager ÒÐ | 52 |
| 4.3. Session Server × U. | 52 |
| 4.3.1. × U P{ | 52 |
| 4.3.2. 3O a | 52 |
| %! 3O a | 52 |
| \@¾° /O a Ð Workload h (.... | 52 |
| \@¾° /O a Ð Serviceh (.... | 53 |
| 4.3.3. Manifest %& × U. | 54 |
| Workload. | 54 |
| Service. | 55 |
| 4.3.4. Server —ž de | 56 |
| 4.4. WAS × U. | 56 |
| 4.4.1. × U P{ | 56 |
| 4.4.2. 3O a | 57 |
| %! 3O a | 57 |
| \@¾° /O a Ð Workload h (.... | 57 |

| | |
|---------------------------------|----|
| \@¾° /O a Ð Serviceh (| 58 |
| 4.4.3. Manifest %& × U | 58 |
| Workload. | 58 |
| Service. | 60 |
| 4.4.4. Server —ž de | 61 |
| 4.5. Embedded WAS × U | 61 |
| 4.5.1. × U P{ | 61 |
| 4.5.2. 3O a | 61 |
| %! 3O a | 61 |
| \@¾° /O a Ð Workload h (| 61 |
| \@¾° /O a Ð Serviceh (| 62 |
| 4.5.3. Manifest %& × U | 62 |
| Workload. | 62 |
| Service. | 64 |
| 4.5.4. Server —ž de | 65 |
| 4.6. Web Server × U | 65 |
| 4.6.1. × U P{ | 65 |
| 4.6.2. 3O a | 65 |
| %! 3O a | 65 |
| \@¾° /O a Ð Workload h (| 65 |
| \@¾° /O a Ð Serviceh (| 66 |
| 4.6.3. Manifest %& × U | 67 |
| Workload. | 67 |
| Service. | 68 |
| 4.6.4. Server —ž de | 69 |
| 5. ECS %& 34 | 70 |
| 5.1. ECS Å- | 70 |
| 5.2. 34 P{ | 70 |
| 5.3. Manager × U | 70 |
| 5.3.1. 3O a | 70 |
| %! 3O a | 70 |
| \@¾° /O a | 71 |
| 5.3.2. Task 3O | 71 |
| Task O: | 71 |
| Volume ¬r | 72 |
| Container ¬r | 72 |
|] ^İ • 3O | 72 |
| Health Check 3O | 73 |
| Volume I ³ | 73 |
| 5.3.3. Service 3O | 73 |
| #{ O: | 73 |
| #{ L M (Service Discovery) 3O | 74 |

| | |
|--------------------------------------|----|
| 5.3.4. Service %» > de | 74 |
| Service ' " de | 74 |
| Task ' " de | 74 |
| 5.4. Session Server × U | 75 |
| 5.4.1. × U P{ | 75 |
| 5.4.2. 3O a | 75 |
| %! 3O a | 75 |
| \@¾° /O a | 75 |
| 5.4.3. Task 3O | 76 |
| Task O: | 76 |
| Container ¬r | 76 |
|] ^İ • 3O | 77 |
| 5.4.4. Service 3O | 77 |
| #{ O: | 77 |
| #{ LM (Service Discovery) 3O | 78 |
| 5.4.5. Service %» > de | 78 |
| Service ' " de | 78 |
| Task ' " de | 79 |
| 5.5. WAS × U | 79 |
| 5.5.1. × U P{ | 79 |
| 5.5.2. 3O a | 79 |
| %! 3O a | 79 |
| \@¾° /O a | 79 |
| 5.5.3. Task 3O | 80 |
| TaskO: | 81 |
| Container ¬r | 81 |
|] ^İ • 3O | 81 |
| 5.5.4. Service 3O | 81 |
| #{ O: | 82 |
| #{ LM (Service Discovery) 3O | 82 |
| 5.5.5. Service %» > de | 82 |
| Service ' " de | 82 |
| Task ' " de | 83 |
| 5.6. Embedded WAS × U | 83 |
| 5.6.1. × U P{ | 83 |
| 5.6.2. 3O a | 83 |
| %! 3O a | 83 |
| \@¾° /O a | 83 |
| 5.6.3. Task 3O | 84 |
| TaskO: | 84 |
| Container ¬r | 85 |
|] ^İ • 3O | 85 |

| | |
|------------------------------------|-----|
| 5.6.4. Service 3O | 85 |
| #{ O: | 85 |
| #{ LM (Service Discovery) 3O | 86 |
| 5.6.5. Service %» > de | 86 |
| Service ' " de | 86 |
| Task ' " de | 87 |
| 5.7. Web Server x U | 87 |
| 5.7.1. xUP{ | 87 |
| 5.7.2. 3O a | 87 |
| %! 3O a | 87 |
| \@¾° /O a | 87 |
| 5.7.3. Task 3O | 88 |
| TaskO: | 89 |
| Container ¬r | 89 |
|] ^İ • 3O | 89 |
| 5.7.4. Service 3O | 89 |
| #{ O: | 90 |
| #{ LM (Service Discovery) 3O | 90 |
| 5.7.5. Service %» > de | 90 |
| Service ' " de | 90 |
| Task ' " de | 91 |
| 6. VM/Host %& 34 | 92 |
| 6.1. 34P{ | 92 |
| 6.2. LENA 34 | 92 |
| 6.3. abci mn | 92 |
| 6.4. Manager 34 | 94 |
| 6.4.1. Manager 34 | 94 |
| 6.4.2. Manager ^ € | 95 |
| 6.5. Node Agent ^ € | 97 |
| 6.5.1. Node Agent ^ € | 97 |
| 6.5.2. Node Agent » ‡ – É de | 98 |
| 6.5.3. Node Agent pl | 98 |
| 6.6. Session Server 34 (WEB UI %&) | 99 |
| 6.6.1. Session Server 34 | 100 |
| 6.6.2. Server ^ € | 100 |
| 6.6.3. Server ŸE | 100 |
| 6.6.4. Server —ž | 100 |
| 6.7. Session Server 34 (CLI %&) | 101 |
| 6.7.1. Session Server 34 | 101 |
| 6.7.2. Session Server ^ € | 103 |
| 6.7.3. Session Server ŸE | 104 |
| 7. B¶ | 106 |

| | |
|--|-----|
| 7.1. LENA s ^a SpecB O; | 106 |
| 7.2. Manager DBØÛ e ´ | 106 |
| 7.3. Manager : ? É † j Ÿ E | 106 |
| 7.4. Manager : admin Q # § Ã % Û | 106 |
| 7.5. LENA 3 4 ± Ç OSØ y î Ž (CentOS % P) | 107 |
| 7.6. LENA Î % \ Q D % r * \$ Ø Û | 108 |
| 7.7. WAS Image OS M ¢ H I | 109 |

Chapter 1. OVERVIEW

! " # \$ Container %& LENA Server' () * % + # , - . Architecture/O- 1 2 3 4 5 6 7
%8. 9. LENA : ; < % = > () 5 6. ?@A B C D E F G \$ () H I J K L M N. 9.

! " # \$ LENA 1.3.1c O; L % P Q D % 8 * N, 9 R S T A ? @ L U V. 9.

¥ LENA for Container W X Y Z / O- 1

[\ @] ^ 5 6. N _ ` a

[Server b c B N _ ` a

¥ LENA for Container 3 4

[Base Image Build

[Kubernetes %& 3 4

[Docker %& 3 4

[ECS %& 3 4

[VM / Host %& 3 4 (Manager, Session Server)

1.1. ! " # \$

LENA\$ Web Server, Application Server, Session Server2 Web Server: Status' de * \$ Node
Agent, Application Server5 3 4 Gf Status Og' EF * \$ Advertiser2 hi H5j EFG\$ kl hi
Cme ManagerD mn o 9.

1.1.1. Server

LENA5# EFG\$ #O: pq\$ Web Server, Application Server, Session Server 3r sr t 9. u
#O: @C\$ Wv 2 T 9.

1. Web Server: ` @H - w5 xy Web Resource' EF. 9. Application Server EF * \$
z @#{ | : Front} ~L • € * • #, , f \ QD Load Balancing > g, ...† f (SSL)' EF * \$
} ~L • €. 9.

2. Application Server: JavaD ‡ n o z @ #{ | ' ^ € / EF . 9.

3. Session Server: Application Server%o ` @H: Š < L b s . 9.

1.1.2. Agent, Advertiser

Node, Server5 3 4 Gf Ef > Ė • Ž • %= L • ' * \$ Agent † 9.

¥ Node Agent

[Web Server ' " Ė • Ž • " † Ž' • I * – Manager5j EF . 9.

¥ Advertiser

[Application Server ' " Ė • Ž • " † Ž' • I * – Manager5j EF . 9.

1.1.3. Manager

Manager\$ Node Agent2 Advertiser' k * – Node2 Server: Ef > Ė • Ž • %= – L
EF * \$ Web Application† 9. 6 ~ \ QD Wv 2 T A %= L EF . 9.

| | |
|-------------------------|---|
| %& | ' (|
| Dashboard | ¥ Server, Service Cluster ™Š ¥ Notification de |
| Server | ¥ System (> i \ Server œ•) —ž /• O/Ÿ E |
| Service Cluster | ¥ Service Cluster —ž /• O/Ÿ E ¥ —ž o Server ž , † j ¢ £ ¥ Service Cluster ¤ ¥ , Revision hi ¥ 3O Template 9(DŠ' k . C I C D ¨ © %= ¥ ª « Terminal > Standard Out/Error Log ¢ £ (Kubernetes5 . V) |
| Resource | ¥ Resource (Database, DataSource, Application,k8s config) —ž /• O/Ÿ E/¢ £ ¥ Resource' ` @* \$ Server ž ¢ £ > -r /E - |
| Diagnostics (£• Ž•) | ¥ Server5 6. † ® ™Š £• Ž• %= ¥ Server5# ¨ ° . Event ¢ £ %= |
| Topology | ¥ SystemB Server mn™Š ¢ £ |
| Admin | ¥ ` @H > ±. hi , ` @H/±. /² J I ³ ¥ ` @H () † j ¢ £ ¥ y† , hi , ™Š ¢ £ > ´ DŠ ¥ Cloud Profile hi |

1.2. Mechanism

LENA\$ Manager' k 7# Web/Application #O' £• Ž• > kI hi * \$ %=L EF. 9. 9µ,
%¶ Host/VM. S:] ^\ ¹ †° A Container5# m» G\$ Server\$ OrchestrationCm5 : 7
^ €† EF G¼, State' r ss ½\$ 9\$ ° † 9.

xy#, %¶ Host/VM] ^5# Agent' k. ^ ¾% Ef/3O hi . 6¿ ÄB Serverr
Container%» ¾° 5 Manager' k 7 3OOg > y† , | ' 9(DŠ Á\$. QD 3OOg'
EF * N, %» o Server: ' " ' Manager' k 7# £• Ž• * \$. QD hi . 9.

ServerÄ: mn- 1D\$ %» ¾ 9(DŠ > Ä%3O %=L ` @* % Ä7# \$ u Server\$
Container¾‡ CommandD Ä@G\$ docker-entrypoint.shr t N, () G\$ Server: ' " ' ; Æ* %
Ä7 Web #O5\$ Node Agentr 34GN Application > Session Server\$? Ço (EÈL Ä@. 9.

WEB-WAS ¨ © ÉÊ5#C %¶ VM/Host] ^S: ¹ †r ¶É. 9. Container: ° n/1î 5 xy IPÍ
Î 1r Î » \e] ^5# Back-End Application Container2: sÐ\ e ¨ /L bs * % Ä7# \$ Load-
balancerr , - * ¼, † \$ Kubernetes: ServiceÍ ECS: Service Discovery, EKS/ECS: ELB c" D
EFGN t 9. %¶ VM/Host] ^5# WEB-WAS' ÑÒ ¨ / * N WEB #Or ÑÒ Load-balancingL
• €ÓÔ . A Container] ^5# \$ EFGs ½N ¥ÖÖ† EF * \$ Service Î 1 (Service Endpoint)D

Reverse Proxy / L E F . 9.

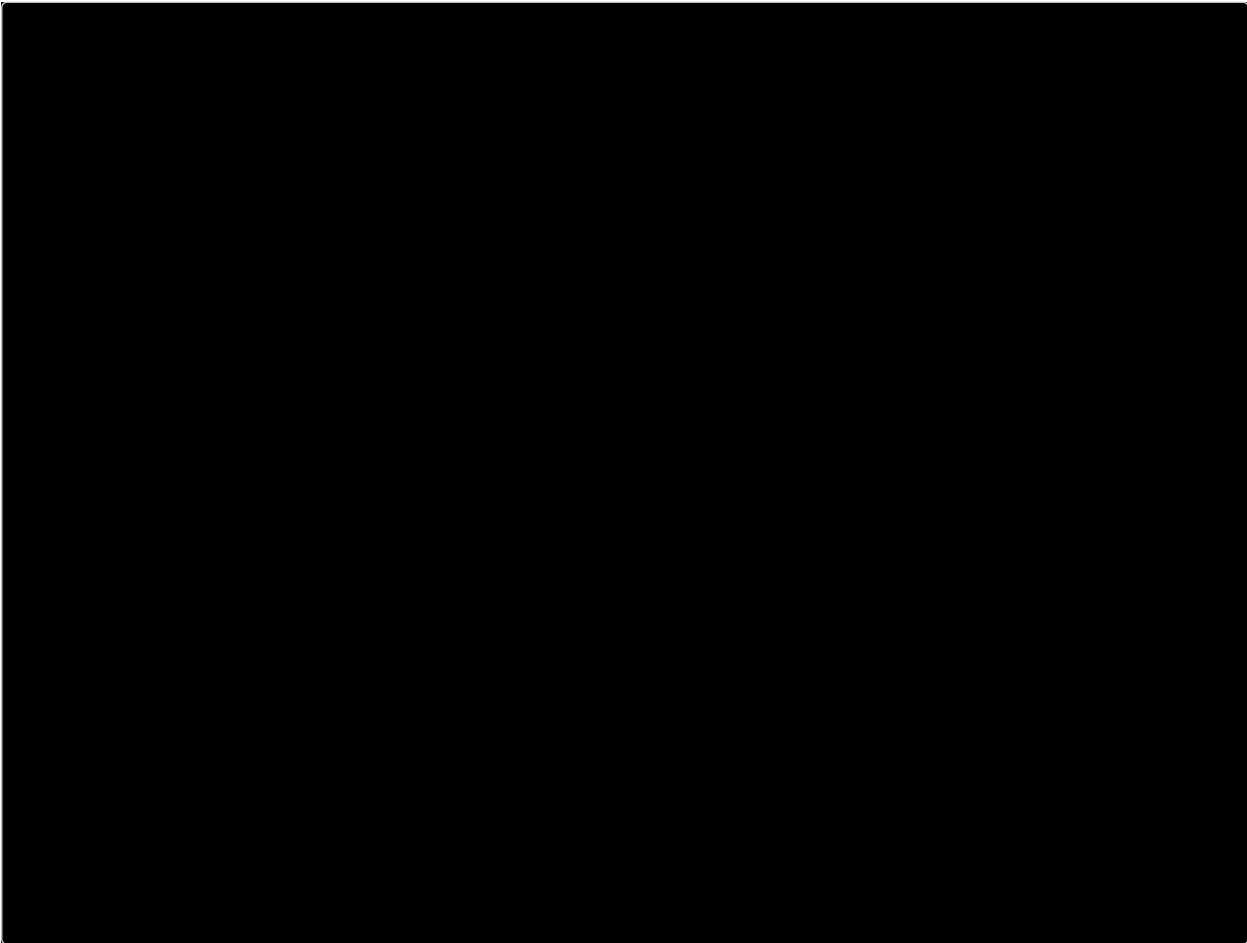


Figure 1. Container %& LENA Server%'' © mΦ (Kubernetes] ^)

| %& | ' (|) * |
|----------------------|---|--------------|
| Application Server | Application Server Instance | |
| Web Server | Web Server Instance | |
| Session Server | Session Server Instance | |
| Manager | # O5 xUG\$ 3OØÛ hi > Server Æ• Ž• %= EF | |
| Manager Repository | Manager () L Ä. ØÛÚÇ Repository, u p 3OOg > DB Og' UVV | ÂÉÚÇ1DÊi r = |
| docker-entrypoint.sh | Container %» ¾5 ^ €G\$ Shell Script 1. %» ¾° 5 3OOg Ã%Û 2. ManagerD ÉŽ 3OOg/y†, 9(D\$ 3. #O %» %=L • €V | |

| | | |
|------------|--|---------------------------|
| %& | ' (|) * |
| Node Agent | Web #O Œ• Ž• " †Ž •I > Manager5j Æ¿, ManagerDEŽ •¿. Ef /3O ÜÝ ^ € | |
| Advertiser | Œ• Ž• " †Ž •I > Manager5j Æ¿ | Application Server5 kI |

Container ()] ^: Þn†Í Eß5 xy Session Server Í Manager' VM/Host] ^5# () ~ • C
t N, LENA Manager\$ Containerc LENA Server àW• y, VM/Host %& LENA Server' hi * \$
%=C UV* N t QáD, 9RS TA WXYZD () â • C t 9.

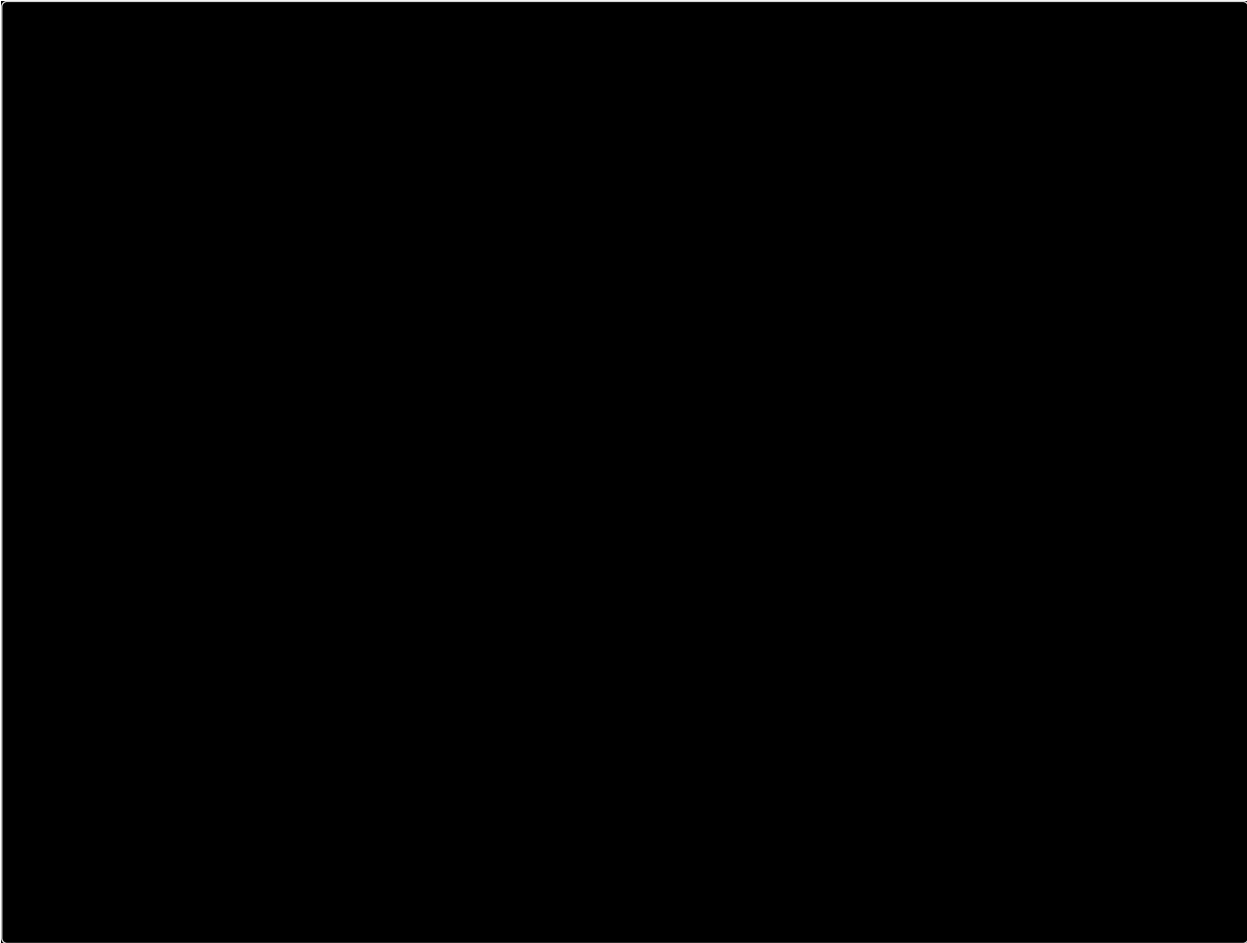


Figure 2. Container -VM/Host āI] ^5#: LENA Server% ¨ © mΦ

1.3. + , Asset

LENA for Container O; 5# \$ 9RS TA Asset EF o 9.

¥ Docker Image : Linux OS + JDK + LENA Server + , - Libraryr UVo Image' Docker Hub'
k 7# EF

- [Web Server : <https://hub.docker.com/r/lenacloud/lena-web>
- [Application Server : <https://hub.docker.com/r/lenacloud/lena-cluster>
- [Session Server : <https://hub.docker.com/r/lenacloud/lena-session>
- [Manager Server : <https://hub.docker.com/r/lenacloud/lena-manager>

¥ Kubernetes Manifest ØÛ : Kubernetes5 34 ¾ , - . Workload / Service / Config Map † %8o
LENA Server× U@ ØÛ

Docker Hub5# EF * \$ Image: SpecA 9RS T9.

| ! - | JVM | OS (Base Image) | . / Heap Memory |
|--------------------|--------------|------------------------|-------------------|
| Application Server | Open JDK 1.8 | ¥ Cent OS 7 (centos:7) | 1.0 GB |
| Web Server | Open JDK 1.8 | ¥ Cent OS 7 (centos:7) | 64MB~256MB(Agent) |
| Session Server | Open JDK 1.8 | ¥ Cent OS 7 (centos:7) | 1.0 GB |
| Manager | Open JDK 1.8 | ¥ Cent OS 7 (centos:7) | 1.0 GB |

1.4. O12 #! 3%

LENA for Container: u #O e | ä | 345 6. å 1 - m` aA 9RS T9.

| ! - | JVM | 4\$ Memory | Image Size(Base Img + 5) | . / ' 6 Memory |
|--------------------|---------|------------|--------------------------|----------------|
| Application Server | JDK 1.8 | 512M | β 900 MB (β 300MB) | 1.25 GB |
| Web Server | JDK 1.8 | 512M | β820 MB (β 300MB) | - |
| Session Server | JDK 1.8 | 512M | β 900 MB (β 300MB) | 1.25 GB |
| Manager | JDK 1.8 | 512M | β 1,000 MB (β 500MB) | 1.25 GB |

u #O 34 ¾ %! , - Memory %PQD 34 G¼, å 1 ` æ ĩ ^ ¾ 3O ç ĩ ^ † , - * 9. Image Size\$ OS + JDK + LENA Server + , - Library ; <' 34. Image è%† 9.

Chapter 2. Architecture 7 6 3%

2.1. 89

; < Architecture : ` / OS 34SOA Wv æéS T 9.



Figure 3. ; < Architecture : ` / OS 34SO

¥ Architecture : ` / O: ¾±A Container ¥ÖÖ: / OS () Container: OS2 JDK' / O* \$
ê†9. †5 x y LENA%& Image' , f* N, ¥ÖÖ5 xë xU. , † / Oo9.

¥ 34\$ WEB/WAS #O2 Manager/Session #O 34. , † 9ì j í €o9. Ù&\QD
Manager/SessionA LENA Image' EFG\$ æ6D ` @*j` GN, WEB/WAS\$ LENA †î s'
%&QD ï DÖñBD , - D* \$ Application / Library—L ¬r 34* \$ ò| ó Base Image'
Build* – Å@* \$. , QD í €o9.

¥ WEB/WAS/Session #O' kI hi * % Ä7#\$ 7' Container' mn* % ; 5 LENA Manager'

Å@* – u ServiceB Service Cluster' ` ; 5 mn* – ô . 9. Service Cluster' ° n* N, Container] ^ 3 O 5 Manager Î 1 2 Service Cluster Og' – r * • Container %» ¾ 5 Template / License 9(D\$ ô 1 r • € G¼, %» ö Manager 5 # %» o Container: Serverr H» – ž GN Ē • Ž • Og' de ~ • t 9.

2.2. Container : ; < = > ? @ * A 3 %

ProviderB 9æ. Container ()] ^† EF GN t QÍ è j EKS/GKE/AKS—L UV* \$ Kubernetes] ^ S Amazon ECS 2 T A Docker %&] ^ 2 r s bc QD Í ÷ f ø • t 9.

Container] ^ B P n ù LENA' () * \$ ") ú L î 4 \$ Î - P n A 9 R S T 9.

1. mn Server% N/W k ¿

† N_` a A ¾ | ð L mn* \$ Server û † Container ()] ^ : ? É N/W 2 Â É N/W 5 Ê ü G f t L ^ ý , ' p k ¿ r = – É 5 6. N_` a † 9. Û & \ Q D P B. N/W E ß † 3 O G f t s ½ Q • outbound k ¿ † r = * ¼, LENA: ^ ý Server ý Manager, WAS ý Session Server: k ¿ † r = * – ô . 9. P ! , VM + Container' ã l * – mn ~ ^ ý ECS: vpc" ñ # è Ē S Z \$ Container 2 VM %: k ¿ † r =. N/W mn † , - * 9.

2. Load Balancing s ^a

VM/Host 2 % i Container \$ • ¾ D ° n / 1 ì â • t Q ¼, † 5 x y IP Î 1 r ï ^ o 9. x y #, Back-End 5 Ä 4. Container ° n / 1 ì ö 5 C s Ð \ e Service' b s * % Ä 7 # Back-End # { | : + & 5 Load-balancer' , - D * j o 9. Kubernetes \$ Load-Balancing L EF * \$ Service' EF * N, ECS \$ ELB " / ' \$ Service Discovery 3 O L k 7 Load-balancing % = L EF * N t 9.

3. Instance) Ð n s ^a (h (- 1 : Session Server, Manager)

NO o Ä • : Container' NO o Î 1 D s Ð \ Q D () * \$ ê L : î * ¼, DBMS 2 T † F b G \$ H ^a # { | ' Container D # { | ~) , - . P n Q D, LENA: mn - 1 ù Manager 2 Session Server: mn 5 , - * 9. Kubernetes: ^ ý \$ StatefulSet x U • L k 7 † P n L EF * N, ECS: ^ ý 5 \$ Replica 1 e Service' x U * – b ` * j () â • t 9.

4. Ä É Volume " / (h (- 1 : Manager)

Containerr State b s' g Ç ~ • * % 5 1 ì / % » † - ° * – C ` @ * \$ Data' s Ð \ Q D b s * % Ä 7 # \$ Ä É Ú Ç 1 (Volume) 5 Og' Ú Ç * – ô . 9. Û & \ Q D DBMS: DB" † Ž Ú Ç † Í + • Ä: Container 5 » Û Application L x U ~) Î D Ä @ o 9. LENA Manager' Container • Q D () ~ ^ ý Ē % » ö 5 C h i Og: Û h n L b s * % Ä 7 # Ä É Volume: " / L , - D . 9.

2.2.1. Container BCD E : ; < = F "

! ö 5 # \$ + 5 # , - . ()] ^ N_` a S h (o Container ¥ Ö Ö: P n L . / ! 9.

Kubernetes

Kubernetes \$ Container Û o Workload 2 Service' h i * % Ä. † , n † t N, d Ç r =. Container Orchestration C m † 9. Kubernetes \$ O i \ Q D \$ Container h i ' Ä. Control Plane S Worker Node D mn G \$ Cluster & Ä D 3 4 G f () o 9. Worker Node 5 \$ > i \ e † ´ F % e Namespace r Ê ü x 4 o 9. Kubernetes # { | \$ Container x U r =. å 1 & Ä e Pod S Pod L æ • Û * – h i * \$ & Ä e Workload, Workload' Network # { | D EF * \$ Service D mn GN, Workload > Service \$ Namespace 5 x 4 o 9.

Kubernetes: Network mΦ\$ Service ÆÈL Ä. Cluster Network† mnGf t N, †' k 7
HostD: Port Open, Load BalancingL EF. 9.



Figure 4. Kubernetes Cluster N/W

Kubernetes: Service\$ Pod 1l 5# ^ €ùe 2¥i 3†< L N/W #{ | D 45* \$ -' Ûo
· 6QD Pod5j Nb. IP Î 1 2 Pod 1l 5 6. &Û DNS ÛL É– * N Load-BalancingL EF. 9.
Kubernetes Service: bc 5\$ 9RS TA 4r sr t 9.

Cluster IP

Kubernetes N/W5# ?É NO IP / Domain Name† ~' GN, †' k 7 Cluster Load Balancing†
† 7f í 9.

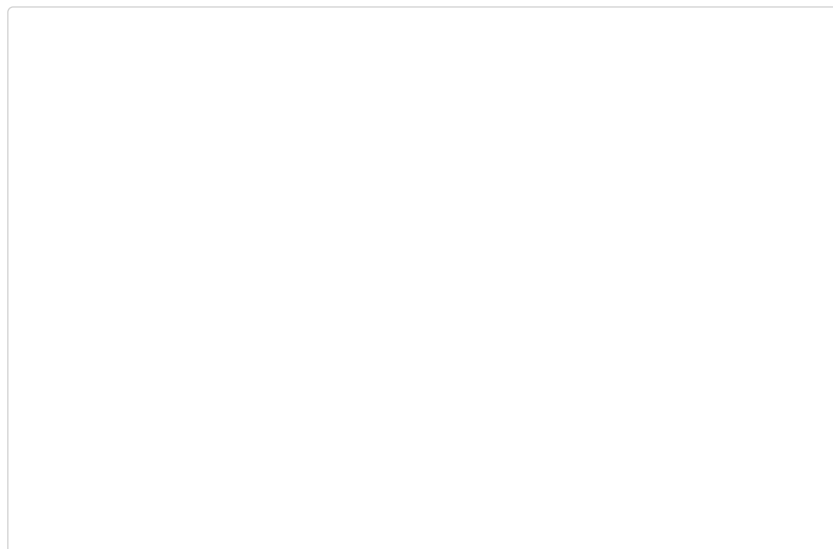


Figure 5. Kubernetes Deployment bc - Cluster Ip

Node Port

Cluster' mn * \$ 8 Node: Port' Container PortD `` / . 9. Node 30000-32767 9Ä:
Portr OpenGN, %! \QD\$: ; *j Portr sOGÍ NO\QDC sO~ • t 9.

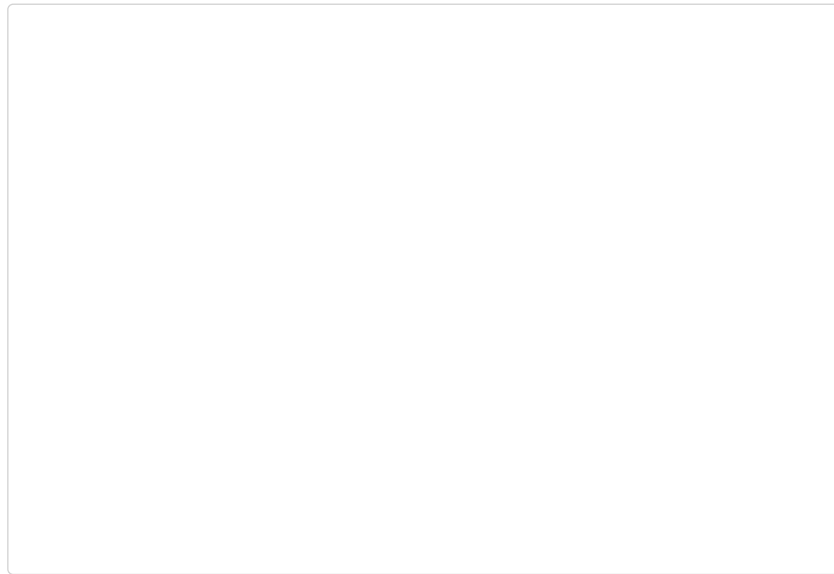


Figure 6. Kubernetes Deployment bc - Node Port

Load Balancer

Node Port' <=VS » 345 Container N/W 5 t \$ LoadBalancer' `` ©* – Service' 45. 9.
EKS2 TA Cloud Service5# \$ Cloud Service Provider5# EF * \$ Load Balancer' ° n* –
`` / . 9.

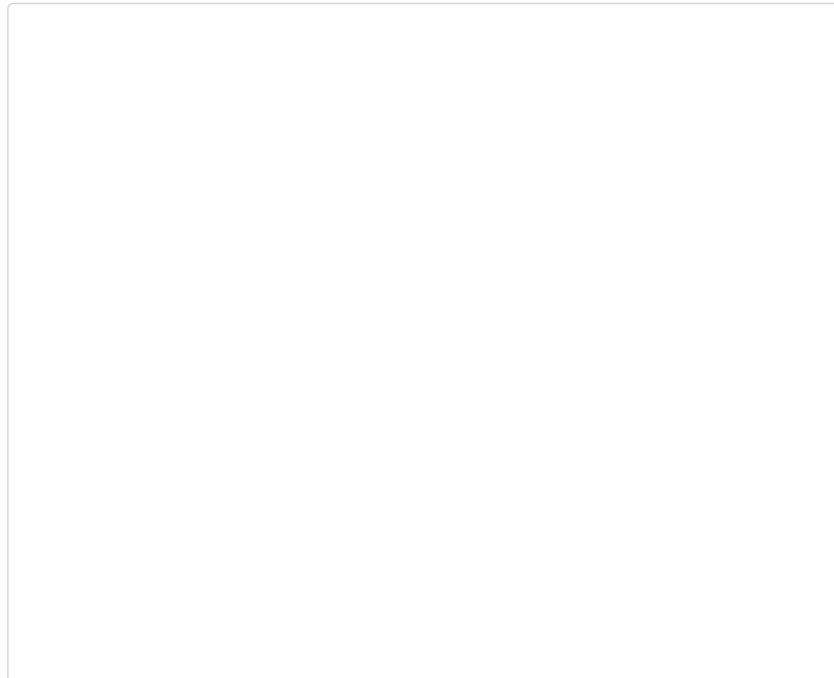


Figure 7. Kubernetes Deployment bc - Load Balancer

Headless

BC: Service Cluster IP * † Domain NameµL k. Load Balancing • €. 9. Pod BD uu:
Domain† sOGN, Stateful Set' †@* \$ ^ý Î D` @o9.



Figure 8. Kubernetes Deployment bc - Headless

Kubernetes\$ 9æ. bc: Container(Pod) xU . 5 L s^a . 9. Û&\ QD Deployment(Replica Set)L ` @* sµ NOo Instance Å• ' , - D* \$ LENA Manager, Session Server5\$ Stateful Set \@† \I * 9.

Replica Set

Node Å• 2 h©* † - wo • µ>: Replica' ° n. 9. Æ8 Pod† » Û. Persistent VolumeL Fb* \$ c" Dmn~ • t 9.

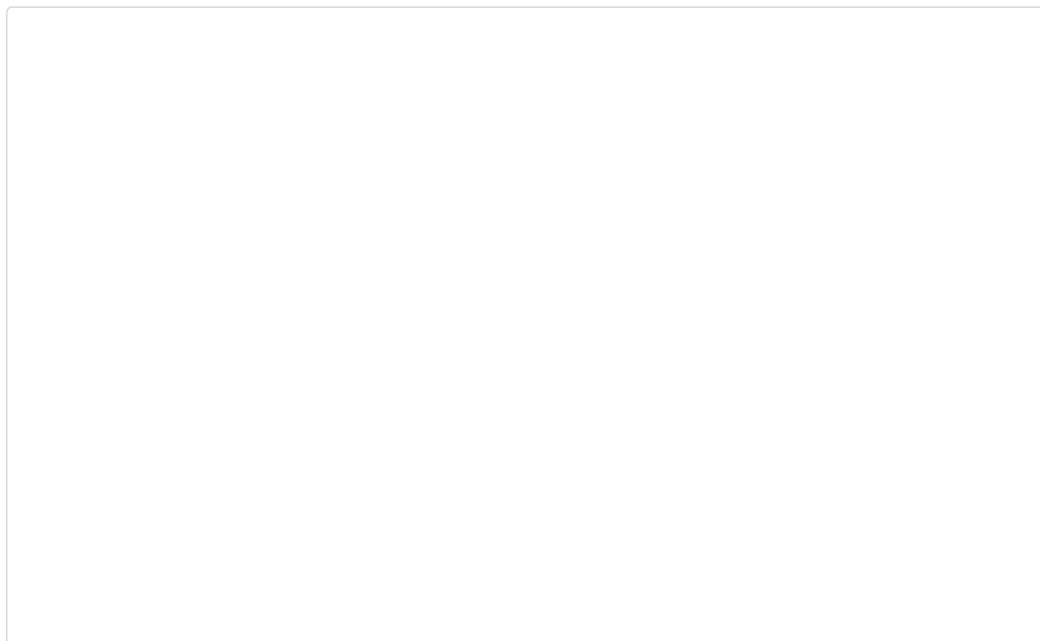


Figure 9. Kubernetes Workload bc - Replica Set

Deployment

Replica SetL Ë° n ~ • t N, Versioning ~ • t 9. Û&\ QD WEB #O, Application #O' xU~) ` @o 9.

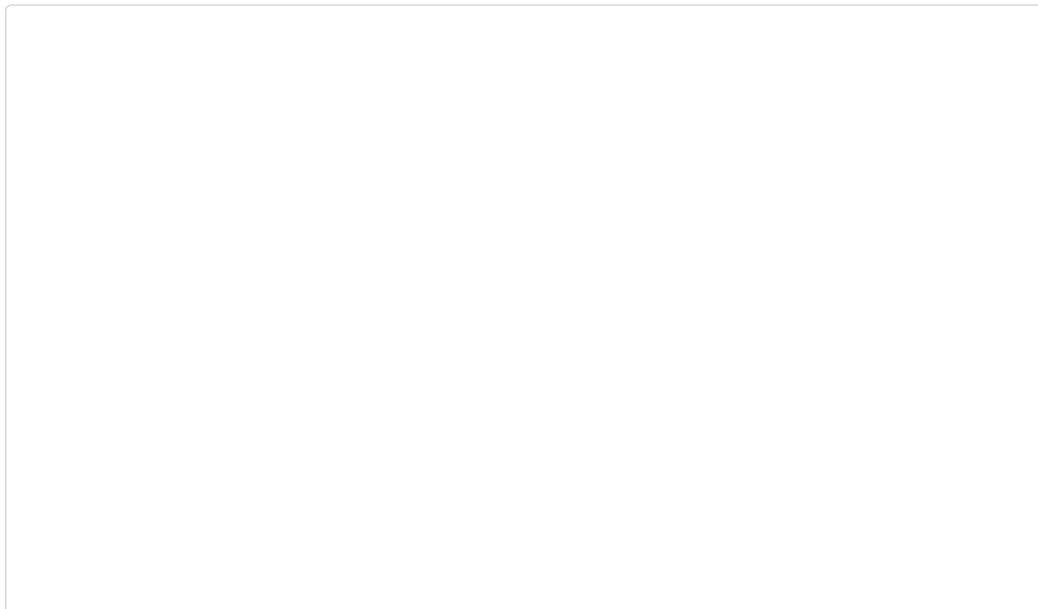


Figure 10. Kubernetes Workload bC - Deployment

Stateful Set

NOo Å• : PodL bS ~ • t N, Pod BD Master / Slave —: ' " çL r ? • t N, Pod BD
 uu: Persistent Volume ~' † r = * 9. ù&\ QD DBMS, Session Server —) Ðn† , - .
 #{ | ' xU~) ` @. 9.

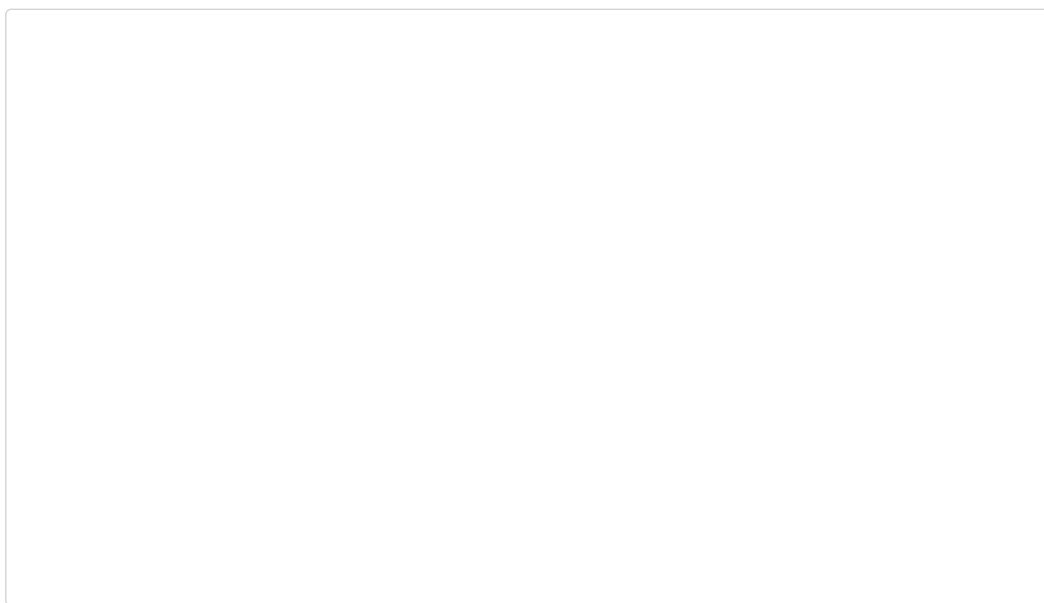


Figure 11. Kubernetes Workload bC - Stateful Set

Daemon Set

Kubernetes Cluster: Worker Node Å• 2 » ù. Pod† Node BD xUGN bso9. ù&\ QD
 Standard OutQD 5j o Log • 1, NodeB Æ• Ž• Og • 1, Ingress' 6ç * \$ Web #O' xU~
 ^ý ` @â • t 9.



Figure 12. Kubernetes Workload bC - Daemon Set

ECS

AWS: ECS\$ Task (Kubernetes Pod2 b`)2 N/W, Replica Set—L 3O~ • t \$ ServiceD mn o9.
 ECS: Service' mn * \$ Task Instance5 6. Load BalancingA 1) ELB • ' \$ 2) Service Discovery
 • QD EF~ • t9. ELB • A Service O: 5# ELB' sO*— 3O~ • t9. Service Discovery
 • A ECS #{ | : Task Instancer ° nG• # Service5 3Oo DNS †@QD Amazon Route 535
 H» —ž*— †' †@. Load BalancingL EF* \$ • †9. ÆÉ ñvA5 : . É* > BC†D
 ' " 5 xy #{ | r dÇG-Í E1GFyC Route 53 p| G) } † âž ' " D bsGáD
 VPC? É5# u #{ | : ' " ' %PQD DNSD `` / † o9. Route 53A Namespace, Task IPB A
 ...H\$ > ‡´ IP+ UñB SRV ...H\$' ° n*— Service5 `` / o9.

ECSÎ - mn - 1

```
% Namespace Ð " I | J † | $ ñvAL yýG~ 6' C²e †@(K: internal, local, corp)L
sO. 9." I | J † | $ #DLMr=*j m™Gf ô * $ #{ | %o: > i \ ^©†9.

% Service Ð #{ | $ " I | J † | „ 5 UVo 2¥i 3†<: Set†9. #{ | 5$ #{ |
e | ä | (Task)r UVGf t9.

% Task Ð Kubernetes: PodS b`. ObjectD &Û ' $ +• Å: Container' *Í : InstanceD
œ• ³ *— hi *¼ Container Instance: Image / ] ^3O / Entry Point— L 3O~ • t9.
```



Figure 13. ECS Service Discovery

Container BCD E F" GH

LENA: () · S h(o u] ^û: PnL Oi * • 9RS T 9.

| Container : ; < = | | 5I N/WJ K | L/BLM | Instance ; N" | * 6 O\$ | 5I Volume P7 |
|-------------------|-----|------------------------------|---|---------------------------------------|---------------------------------------|----------------------|
| Kubernetes %& | Û& | r = | Service L/BS ^a | s ^a | s ^a | s ^a |
| | EKS | VPC? k¿ | Service, ELB'' / | s ^a | s ^a | s ^a |
| Docker %& | ECS | VPC ? k¿ (VPC N/W Æ\$) | ELB, Service Discovery s ^a | s ^a (Service Replica=1) | s ^a (Service Discovery) | s ^a (EFS) |

2.2.2. LENA Server QRE : ; ST

Ä Pn5 x y Container] ^S LENA ServerbcB s^a r =. () · S A 9RS T 9.

| Container : ; < = | | WEB Server | WAS | Session Server | Manager |
|-------------------|-----|------------|-----------|----------------------------------|----------------------------------|
| Kubernetes %& | Û& | Container | Container | Container (statefulset), VM/Host | Container (statefulset), VM/Host |
| Docker %& | ECS | Container | Container | VM, Container | VM, Container |

2.3. , J * A3%

9RA Server bc5 h©* † Fk\QD N_7ô ~ - 1†9.

2.3.1. OS

LENA Image %PQD 9RS TA OS' ` @. 9. † \$ LENA ImageOS¼ ` @G\$ Base Image† 9.

| LENA Image +, OS | LENA ImageU Base Image |
|------------------|------------------------|
| Cent OS 7 | centos:7 |



Cent OS 8, Ubuntu, Debian — P OS' ` @~ ^ý LENA %8sª L k 7# Base Image' Ě° n7ô . 9.

2.3.2. JDK

LENA Image' %PQD OS%! JDK 1.8 (yum / apt-getQD 34) ' \$ Adopt Open JDK 1.8L 34 ` @. 9. 34 QXs >] ^İ • \$ 9RS T9.

| OS | JDK ' (|
|-----------|---|
| Cent OS 7 | ¥ 34 QXs : java-1.8.0-openjdk-devel.x86_64 ¥ JAVA_HOME : /usr/lib/jvm/java |

P JDK' ` @~ ^ý, Project@ Base Image° n¼5 34r r =* Í JAVA_HOME: Path\$ r - \ %¶S » Û*j 34* \$ êL ±N. 9.



\$JAVA_HOME] ^İ • çA u Server: env.sh (manager: ^ý env-manager.sh), LENA 34 Og ({LENA_HOME}/etc/info/java-home.info)5 % ÚÇGf t QáD, JDK Ě34¼5\$ \$JAVA_HOME5 RCž %¶ ØÛ Og • O† , - * 9.

2.3.3. VW User

LENA Image %P ^€ User\$ ſrootſ †9. †' İ ^* NH * \$ ^ý LENA Image' İ ^* – ô * ¼ LENA %8sª L - w* – İ ^* – ô . 9.

2.3.4. Library

LENA: Image5 34Gf t \$ Library\$ 9RS T9.

| Library | XY | OSE Z X[I | Z X Server |
|------------------|--|-----------------|------------|
| net-tools | wget— N/W bSi T | (F k \ @) | (F k \ @) |
| hostname | Hostname d e @ | CentOS | (F k \ @) |
| initscript | Service (Daemon) m» @ | CentOS | (F k \ @) |
| procps | Process h (bSi T | CentOS | (F k \ @) |
| unzip | Server 3OØÛ UE 7 E @ | (F k \ @) | (F k \ @) |
| file | File UV ° L @ | (F k \ @) | (F k \ @) |
| curl | ØÛ 9(D\$ @ | Ubuntu / Debian | (F k \ @) |
| cronie-noanacron | Crontab m» @ | CentOS | (F k \ @) |
| logrotate | WEB/WAS: File Log Rotate Z i @ | (F k \ @) | (F k \ @) |
| libxml2-utils | XML Validation @ (License ØÛ Validation) | Debian | (F k \ @) |
| locales | Locale 3O @ | Ubuntu / Debian | (F k \ @) |
| libapr1 | Web Server ` @ Library | Ubuntu / Debian | (F k \ @) |
| libaprutil1 | Web Server ` @ Library | Ubuntu / Debian | (F k \ @) |
| tzdata | Time Zone 3O | Ubuntu / Debian | (F k \ @) |
| openssl | Web / WAS ` @ Library | (F k \ @) | (F k \ @) |
| awscli | Manager5# EKS API p5 @ | (F k \ @) | Manager |
| pip | | | |

2.4. Server QRE * A 3% Ð Manager

2.4.1. \]

Manager: xU\$ Container ' \$ VM/Host xU ÆW r = * N, æ · , L \ @ * % Ä. Eß` aA 9RS T 9.

1. NO Domain ' \$ IP Î 1 ~'

Container5 34o Serverr 3O ØÛ/y t, | 9(D\$ > Æ• Ž• Og' Ä¿ * áD, sÐ\ e
#{ | ' Ä7# ĚÉG/Ě° n ö5C NOo Î 1 D #{ | r Gf ô V.

2. Server ý Manager %o N/Wk¿

Manager 9(D\$ #{ | ` @ > Æ• Ž• Og EFL Ä. &· ú k¿ t , - . %! 3O %PQD
TCP Port 7700, UDP / TCP Port 16100 ØÐ† X@Gf ô V

3. Persistent Volume

Container5 34o Manager: ^ý È%» ö5C #{ | `` ÐnL EF*% Ä7 DB, 3OOg, Æ• Ž• " †Ž—L ÚÇ~ • † \$ ÂÉ Volume† , - . NFS, EBS Disk, Local Node Disk—L Å@. Persistent Volume' dg* N, Manager Container5 ~' * - ` @

4. Instance) Ðn gÇ

Manager: Instance\$ #{ | `` Ðn EFL Ä7 Instance) ÐnL gÇÁWô . 9. Û& VM /Host\$ %! \QD) ÐnL gÇ*sµ, Container] ^5#\$ Kubernetes: StatefulSetZ\$) ÐnL gÇ* \$· , QD xUGf ô V

| \] ST | + ^(_#) 3% |
|---------------|--|
| Container x U | ¥ Persistent Volume ¥ Server ý Manager % N/W k¿ (TCP Port 7700, UDP/TCP Port 16100) ¥ NO C² e ' \$ NO IP ~' ¥ Container) Ðn gÇ |
| VM/Host 34 | ¥ Server ý Manager % N/W k¿ (TCP Port 7700, UDP/TCP Port 16100) ¥ NO C² e ' \$ NO IP ~' |

2.4.2. 3`

Manager Server' () *% Ä7# , - . ` æA 9RS T9.

Memory

Manager Server: Heap Memory Size\$ å1 512Mbyte' , - D * N Image5\$ 9RS T† %! 3O† Gf t9.

¥ Heap Memory : 1024 Mbyte

¥ Metaspace Memory : 256 Mbyte

†' İ ^* NH * • 9R] ^İ • ' 3O* - ¢O . 9.

¥ LENA_JVM_HEAP_SIZE

¥ LENA_JVM_METASPACE_SIZE

!

Ä] ^ İ • : ÇA MByte&Ä† ¼ &§¾ YH+ôm c " : UV (K : 1024m) c , QD sOGf ô . 9. UV† ZÜ4 * • \@Gs ½\$9.

Disk

LENA Image %POD ` @G\$ Image è%\$ B 1,000 MbyteD † \$ OS + JDK + LENA + Library: [I QD Image: ' Ä Layer @\] s UV. @\ †9.

-%5 -r \QD N_7ô ~ Disk@\A 1) Manager Log ØÛ @\S 2) Repository (DB > ØÛÚÇ1) †9. Manager5# ` @G\$ Repository : Ä4\$ \${LENA_HOME}/repository† N, 5GB OC: @\L , - D . 9. æi N, Container5 34â ^ý5\$ † Repository\$ Persistent Volume5 `` / * - ContainerÄÉ5 ÚÇ* - ô . 9.



Manager5# `@G\$ Repository : Ä4\$ \${LENA_HOME}/repository†N,
Container Ê%»¾45C hi "†Ž: sĐnL Ä7 Container5 34â ^ý5\$
Persistent Volume5 ``/* – ContainerÄÉ5 ÚÇ*%' ±N. 9.

2.4.3. ' 6

abcd

- 1. " ñ#èÎ 1
Manager\$ &§¾ NO Domain ' \$ IPÎ 1' ~' * – ô . 9. (! " # xUMÇ)
- 2. #{ | Uñ
Manager: #{ | Uñ\$ 9RS T† NOGf t9.
[Http (TCP) Port 7700: kI hi #{ | > Rest API #{ | EF
[UDP Port 16100: Ė• Ž• "†Ž • 1
[TCP Port 16100: Thread / Service Dump "†Ž ° n / • 1

ÊUñ\$ LENA Manager Image5 NOGf tN, Ĩ ^L *NH *\$ ^ý
LENA Image Ĩ ^g9\$ Container: %! 3O ` ' 5 xy Port MappingL
Ĩ ^~ êL ±N. 9.



LENA Manager: Service Uñ Ĩ ^A P Server2: ``» Port Ĩ ^L : î *¼,
Ĩ ^¾45\$ ``» Gf t\$ Ė8 Server: 3OL Ĩ ^/Ë¾±L , - D . 9.

<=ef

Manager Container5 \@ r =. Î -] ^İ • \$ 9RS T9.

| <=ef | ' (| . / g | e=hi |
|-----------------------------|---|---------|------|
| LENA_JVM_HEAP_SIZE | ¥ Heap Memory è% sO | 1024m | ^ |
| LENA_JVM_METASPACE_SIZE | ¥ Metaspace Memoryè% | 256m | ^ |
| LENA_MANAGER_DOMAIN_ENABLED | ¥ Domain Name ÅnÛ – É ¥ 0Y0' \$ 0N0 | Y | ^ |
| LENA_MANAGER_ADDRESS | ¥ ÄÉD 45G\$ (Serverû† e , * \$) Manager Î 1 ¥ c , : IP / DomainÎ 1 :#{ Uñ K) Kubernetes: ^ý : Service Domain | | ^ |
| JAVA_DOMAIN_CACHE_TTL | ¥ Domain Î 1 Cache ¾4% (Ä) | 3 | ^ |
| LENA_SERVER_TYPE | ¥ Server bc | manager | – |

| | | | |
|------------------|-------------------------------|---------------------|-----------|
| < = e f | ' (| . / g | e = hi |
| LENA_HOME | ¥ LENA 3 4 Ä 4 | /usr/local/l ena | _ |
| LENA_JVM_OPTIONS | ¥ ` @H O: JVM OPTION | | ^ |
| LENA_USER | ¥ Manager %» 5 ` @~ OS ` @H©O | root | ^ |
| LENA_USER_GROUP | ¥ Manager %» 5 ` @~ OS ` @Hœ• | root | ^ |



JAVA_DOMAIN_CACHE_TTL ç† 3OG` L ¾
\${JAVA_HOME}/jre/lib/security/java.security ØÙ: networkaddress.cache.ttl ç L
İ ^ . 9.

Directory ! j

LENA Image %P %! 3 4 Ä 4\$ '/usr/local/lena' † N, œ * Ä m¢\$ 9RS T 9.

| | | |
|--|---|--------------------------------|
| k l mn (\${LENA_HOME} o p) | ' (|) * |
| bin | Manager: Start/Stop scripts | |
| depot | 3 4' Ä. Local Repository | |
| etc | %P ² P Og > 3O ØÙ | |
| license | License Og' hi * \$ a b c i | |
| logs d lena-manager | Dœ ØÙ ÚÇ1 Home Manager LogØÙ ÚÇ1 | |
| modules d lena-manager | LENA E F ÆÈ: ÚÇ1 Home lena-manager ^ €5 , - . ÆÈ† Ä4* \$ ^ D | |
| repository d backup d config d container d database d license d monitoringDB d resource d template | Manager" † Ž ÚÇ1 Home e ´ " † Ž ÚÇ1 Manager 3OOg ÚÇ1 Container 3OOg ÚÇ1 Manager " † Ž f † ÚÇ1 y † , ØÙ ÚÇ1 Æ• Ž• " † Ž ÚÇ1 Resource ´ D\$ ØÙ ÚÇ1 Server 3O Template ÚÇ1 | Container5# ÂÉ øg ` @ ±N |
| tmp | I ¾a b c i | |

Log & Dump q r

Log > Dump\$ Standard Out / ErrorD 5i * \$ 'console' · ¸ S FileD 5i * \$ 'file' · ¸ † sª GN,
] ^İ • 'LOG_OUTPUT_TYPEØ: ç L 'console' ' \$ 'fileØD 3O VQD# 5i · ¸ L ;] ~ • t 9.

1. Console 5i

Ù&\QD Container] ^5# h† Å@G\$. 9. Manager: Application Log, Access Log, GC Log' ÆW Standard OutQD 5j . 9. Docker5 3Oo Log Driver5 : 7 Node(Host): sOÄ4 (%! Ä4 : /var/lib/docker/containers/[container-id]/[container-id]-json.log)5 ÚÇG- Í FluentD2 TA Log Aggregator5 : 7 • 1 / ÚÇ* – kl hi ' ~ • t 9.

2. ØÙ 5j
ØÙ 5j 3O ¾ LogØÙ > Dump ØÙA \${LENA_HOME}/logs/lena-manager* Ä5 Daily Rolling . 9. QD ÚÇo 9. ÚÇo ØÙA %! 3O5 xy å p %ž os30Ù t' ^So LogØÙL I Û ÿÉ. 9.
† . 9. A Manager VM/Host] ^5# () G- Í Manager Log' BCD • 1 * s ½\$] ^5# ` @~ • t 9.

Health Check

Health Check\$ Kubernetes' %PQD 3Ü. 9.

Health Check\$ %» ¾ Container: #{ | P{ – É' i &* \$ 1) Readiness Probe2 (@ù O' \e #{ | – É' <è * \$ 2) Liveness Probe, œi N Application ¾† – É' i &* \$3) Startup Prober t 9.

| ! - | XY |
|-------------------|---|
| Readiness (Probe) | %» ¾°, BC† Dr - wL Zi ~ P{ r G` \$s – É |
| Liveness (Probe) | () ¾°, BC† Dr O' » ‡ ùes – É |
| Startup (Probe) | BC† D?: 2¥i 3† < † ¾† G` \$s – É |

Check . 9. (Action)5\$ 9R 3r s bc† t 9.

| ! - (Action) | ST |
|--------------------------------|--|
| TCP Socket (TCPSocketAction) | Port k¿ – É D Health Check |
| Http URL Query (HTTPGetAction) | URL p5 / S H§ D Health Check |
| Exec ^ € (ExecAction) | Container ? É: ÜÝf ^ € QD Health Check |

Manager' Health Check * \$. 9. A URL <è . 9. L ` @* ¼, Readiness2 Liveness Probe' \@. 9. %! 3OA 9RS T 9.

1. Readiness Check

```
[ httpGet : path /lena, port 7700
[ initialDelaySeconds : 5
[ periodSeconds : 5
```

2. Liveness Check

```
[ httpGet : path /lena, port 7700
[ initialDelaySeconds : 20
[ periodSeconds : 5
```

2.5. Server QRE * A 3% Ð Session Server

2.5.1. \]

Session Server: xU\$ Container ' \$ VM/Host xU ŒW r =* N, æ · , L \@* % Ä. Eß` aA 9RS T9.

1. NO Domain ' \$ IP Î 1 ~'
Session Server \$ 2Ä: Container 5 Cluster D mno9. u Session Server\$ ' 6j Session Server : Domain / IP' Mirror Server OgD e , * – Session Og' »%Û * áD, sÐ\ e #{ | ' Ä7# ĚÉG/Ě° n ö5C NOo Î 1D #{ | r Gf ô V.
2. Instance) Ðn gÇ
Session: Instance\$ #{ | `` Ðn EFL Ä7 Instance) ÐnL gÇÁWô . 9. Û& VM /Host\$ %! \QD) ÐnL gÇ*sµ, Container] ^5#\$ Kubernetes: StatefulSetZ\$) ÐnL gÇ* \$ · , QD xUGf ô V

| | |
|--------------|--|
| \] ST | + ^(_#) 3% |
| Container xU | ¥ NO C² e ' \$ NO IP ~' ¥ Container) Ðn gÇ |
| VM/Host 34 | ¥ NO C² e ' \$ NO IP ~' |

2.5.2. 3`

Session Server' () * % Ä7# , - . ` æA 9RS T9.

Memory

Session Server: Heap Memory Size\$ å1 1024Mbyte' , - D * N Image5\$ 9RS T† %! 3O† Gf t 9.

¥ Heap Memory : 1024 Mbyte

†' İ ^* NH * • 9R] ^İ •' 3O* – ¢O . 9.

¥ LENA_JVM_HEAP_SIZE

!

Ä] ^ İ • : çA MByte&Ä†¼ &§¾ YH+îmîc " : UV (K : 1024m) c , QD sOGf ô . 9. UV† ZÛ4 * • \@Gs ½\$9.

Disk

LENA Image %PQD ` @G\$ Image è%\$ ß 800 MbyteD † \$ OS + JDK
LENA + Library: [I QD Image: ' Ä Layer @\] s UV. @\ †9.

– %5 –r \QD N_7ô ~ Disk@\A Session Server Log ØÛ †9.

' 6

abcd

1. " ñ#è Î 1
Session Server\$ &§¾ NO Domain ' \$ IP Î 1' ~' * – ô . 9. (! " # xU· , M¢) WAS

> Secondary Session Server5# NOo Î 1 D Session Ogr k¿ o9.

2. #{ | Uñ
Session Server: #{ | Uñ\$ 9RS T† NOGf t9.

[Http (TCP) Port 5180*: Session ¢E > hi Uñ
Ä Uñ\$ LENA Image5 NOGf tN, Ĩ ^L *NH *\$ ^ý LENA Image Ĩ ^g9\$
Container: %! 3O ` ' 5 x y Port MappingL Ĩ ^ ~ êL ±N. 9.

!

Session Server: Service Uñ Ĩ ^A P Server2: `` » Port Ĩ ^L : î *¼,
Ĩ ^¾45\$ `` » Gf t \$ Æ8 Server: 3O Ĩ ^ / Ê¾±L , - D . 9.

<=ef

Session Container5 \@ r =. Î -] ^İ • \$ 9RS T9.

| <=ef | ' (| . / g | e = hi |
|-------------------------------|--|---------|-----------|
| LENA_JVM_HEAP_SIZE | ¥ Heap Memory è% sO | 1024m | ^ |
| LENA_MANAGER_ADD RESS | ¥ ÃÉD 45G\$ (Serverû† e_ * \$) Manager Î 1 ¥ c_ : IP / DomainÎ 1 : #{ Uñ K) Kubernetes: ^ý : Service Domain | | ^ |
| LENA_CONFIG_TEMPL ATE_ID | ¥ Service Cluster Ü : Revision No | | ^ |
| JAVA_DOMAIN_CACHE _TTL | ¥ Domain Î 1 Cache ¾% (Ã) | 0 | ^ |
| LENA_SESSION_0_AD DRESS | ¥ Primary Session #O Î 1, StatefulSet3OS Ü4Gf ô V | | ^ |
| LENA_SESSION_1_AD DRESS | ¥ Secondary Session #O Î 1, StatefulSet3OS Ü4Gf ô V | | ^ |
| LENA_SECONDARY_SE SSION_NO | ¥ LENA_SESSION_0_ADDRESS / LENA_SESSION_1_ADDRESS ù mirror #OD ` @~ Og , f (0,1 µ k j r =) | | ^ |
| LENA_SESSION_EXPIR E_SEC | ¥ Session µI ¾% (Ã) | 1800 | ^ |
| LENA_CONFIG_SHARE _SESSION | ¥ Application %o Session F b – É ¥ 0Y0' \$ 0N0 Ç X@ | N | ^ |
| LENA_SERVER_TYPE | ¥ Server bc | session | — |

| < = e f | ' (| . / g | e = hi |
|------------------|---|---------|-----------|
| LENA_HOME | ¥ LENA 3 4 Ä 4 ¥ %! Ç : /usr/local/lena | (3 ÜMÇ) | – |
| LENA_SERVER_HOME | ¥ Session Server 3 4 Ä 4 ¥ %! Ç : /usr/local/lena/server/sessionServer | (3 ÜMÇ) | X |
| LOG_OUTPUT_TYPE | ¥ Dœ 5 j · ¸ (file/console) | console | ^ |
| LENA_AGENT_RUN | ¥ LENA Agent %» – É | N | ^ |
| LENA_USER | ¥ Manager %» 5 ` @ ~ OS ` @H©O | root | ^ |
| LENA_USER_GROUP | ¥ Manager %» 5 ` @ ~ OS ` @Hœ• | root | ^ |



JAVA_DOMAIN_CACHE_TTL † 30G` L ¾
\${JAVA_HOME}/jre/lib/security/java.security ØÛ: networkaddress.cache.ttl Ç L
İ ^ . 9.

Directory ! j

LENA Image %P %! 3 4 Ä 4 \$ Øusr/local/lenaØ † N, œ * Ä mÇ \$ 9RS T 9.

| k l mn ({LENA_HOME} op) | ' (|) * |
|--|---|-------|
| bin | Session Server: Start/Stop scripts | î ` @ |
| depot | 3 4' Ä. Local Repository | î ` @ |
| etc | %P ² P Og > 3 O ØÛ | |
| license | License' hi * \$ a b c i | |
| modules | LENA E F ÆÈ: ÚÇ1 Home | |
| servers/sessionServer dlib dlogs | Session Server 3 4 Ä 4 (\${LENA_SERVER_HOME}) Session Server Library ÚÇ1 Log ØÛ ÚÇ1 | |
| tmp | I ¾a b c i | |

Log

Session Server\$ File Log 5 j µ EF. 9. 5 j Ä 4 \$ \${LENA_SERVER_HOME}/logs a b c i † ¼
ØÛÜ c , A lena-sessionServer-YYYYMMDD.log QD I Û Log ØÛ† ° n o 9

Health Check

Health Check: %! ?@A ! " # [Manger Health Check](#) É Ê L MΦ. 9.

Kubernetes %P Session Server' Health Check * \$. A Command Exec (ExecAction) . † ¼, \${LENA_SERVER_HOME}/ health.sh' p5. 9.

1. Readiness Check

```
[ exec : ${LENA_SERVER_HOME}/ health.sh
[ initialDelaySeconds : 20
```

2. Liveness Check

```
[ exec : ${LENA_SERVER_HOME}/ health.sh
[ initialDelaySeconds : 30
[ periodSeconds : 5
```

2.6. Server QRE * A 3% Ð WAS

2.6.1. \]

WAS\$ Container5 × UG¼, 3O5 × y Container ¥ÖÖ (Kubernetes, ECS—)5 &Û ' \$ + • Å: Instancer » ¾5 × Uo9.

1. Service ©m

WAS\$ &Û/+• Å: ContainerD × UGN, †' ÂÉÍ Front-End5 Service* % Ä7# \$ + &5 L/B} ~L • €* \$ Service' × 4* \$ ê† Û&\e . † 9. Kubernetes: ^ý5\$ 34o Node: pO PortD #{ | * \$ NodePort, ÂÉ L/B' Å@* \$ LoadBalancer, ?É NO IP' sO* \$ ClusterIp: #{ | bc† EFGN tQ¼, ECS: ^ý5\$ ALB' sO* \$. † t9. ` ; 5 ApplicationL fn. . QD Service~s ` ; /O† , - *9.

2. Instance • (Replica)

&Û Service' * \$ + • Å: WAS: • \$ É* 5 × y rĭ \†Í , Ã% %» 7ô ~ Instance Å• ' ` ; 5 O: *- , ×U 3O5 &) *-ô . 9.

3. Service Mapping

ECS: ^ý5\$ Service5# ÑÒ L/B' sO~ • t sµ, Kubernetes: ^ý5\$ Key-ValueD O: o labelL %PQD Mapping oL O: *- Service2 I ³. 9. ; < ¾| æ ' 5# ù+* N, () 5 j i . Mapping %PL • p*- ×U 3O5 &) *-ô . 9.

2.6.2. 3`

WAS' () * % Ä7# , - . ` æA 9RS T† 9.

Memory

WAS: Heap Memory Size\$ å1 512Mbyte' , - D * N Image5\$ 9RS T† %! 3O† Gf t9.

¥ Heap Memory : 1024 Mbyte

¥ Metaspace Memory : 128 Mbyte

†' ĩ ^* NH * • 9R] ^ĭ • ' 3O*- ΦO . 9.

¥ LENA_JVM_HEAP_SIZE

¥ LENA_JVM_METASPACE_SIZE

!

Ä] ^ Ĩ • : ¸ A MByte&Ä†¼ &§¾ YH+ĭmĭc" : UV (K : 1024m) c , QD s OGf ô . 9. UV† Z Û4 * • \ @Gs ½\$9.

Disk

LENA Image %PQD ` @G\$ Image è%\$ B 900 MbyteD †\$ OS + JDK + LENA + Library:
[I QD Image: ' Ä Layer @\] s UV. @\†9.
() 5 , - . -r\ e Disk\$ Log' ØÛ . , QD ÚÇ~) Log@\S Application 1 | ØÛ
(Artifact): @\L N_ * – üO. 9.

2.6.3. ' 6

abcd

1. " ñ#è Î 1
WAS: " ñ#è Î 1\$ P B. E B† * 9. N/W' 5# Manager2 Session Server' \ s D * \$
&· ú k ç r = * C Ž × 4 7 ô . 9.
2. #{ | Uñ
WAS: #{ | Uñ\$ 9RS T† NOGf t 9.

[HTTP #{ | Port : 8180

Ä Uñ\$ LENA Image5 NOGf t N, ÄÉ #{ | Ĩ ^L * NH * \$ ^ý LENA Image Ĩ ^ g 9\$
Container: %! ` ' 5 xy Port qer 3OL k 7 Ĩ ^ ~ êL ±N. 9.

<=ef

WAS Container5 \ @ r =. Î -] ^ Ĩ • \$ 9RS T 9.

| <=ef | ' (| . / g | e=hi |
|------------------------------|---|-------|------|
| LENA_SERVICE_PORT | ¥ WAS #{ Port | 8180 | ^ |
| LENA_JVM_HEAP_SIZE | ¥ Heap Memory è% s O | 1024m | ^ |
| LENA_JVM_METASPACE_SIZE | ¥ Metaspace Memoryè% | 128m | ^ |
| LENA_JVM_OPTIONS | ¥ ` @H O: JVM OPTION | | ^ |
| LENA_MANAGER_ADDRESS | ¥ Manager Î 1 ¥ c , : IP / DomainÎ 1 : #{ Uñ | | ^ |
| LENA_MANAGER_MONITORING_PORT | ¥ Manager Ĳ • Ž • Port Og | 16100 | ^ |
| LENA_MANAGER_KEY | ¥ Manager Open API ÒĐ c s | | ^ |

| < = e f | ' (| . / g | e = hi |
|-----------------------------------|---|--|-----------|
| LENA_CONFIG_TEMPL ATE_DOWNLOAD | ¥ ManagerDÉ Ž 3O ØÙ 9(D\$ – É ¥ X@Ç : Y ' \$ N | | ^ |
| LENA_CONFIG_TEMPL ATE_ID | ¥ 3O ØÙ ID ¥ c , : Service Cluster Ü:Revision t p | | ^ |
| LENA_LICENSE_DOWN LOAD_URL | ¥ License 9(D\$ URL ¥ k i Ç : manager ' \$ Nu` gbLicense9(D\$ URI | manager | ^ |
| LENA_CONTRACT_CO DE | ¥ License - S h(o ©B H\$D v p Û o Ç I . | | ^ |
| JAVA_DOMAIN_CACHE _TTL | ¥ Domain Î 1 Cache ¾% (Ã) | 3 | ^ |
| LOG_OUTPUT_TYPE | ¥ LOG 5i bc ¥ X@ Ç : console ' \$ file | console | ^ |
| LENA_LOG_OUTPUT_DI R | ¥ Log ØÙ ° n Ä4 | /usr/local/l ena/server s/appServ er/logs | ^ |
| LENA_DUMP_OUTPUT_ DIR | ¥ Dump ØÙ ° n Ä4 | | ^ |
| LENA_SERVER_TYPE | ¥ Server bc | WAS | — |
| LENA_HOME | ¥ LENA 34 Home ¥ Ç : /usr/local/lena | (3ÜMΦ) | — |
| LENA_SERVER_HOME | ¥ LENA #O 34 Ä4 ¥ Ç : /usr/local/lena/servers/appServer | (3ÜMΦ) | — |
| LENA_SERVICE_ENDP OINT | ¥ WASr Ð. #{ : Î 1 | | ^ |
| LENA_AGENT_RUN | ¥ LENA Agent %» – É | N | ^ |
| LENA_USER | ¥ Manager %» 5 ` @~ OS ` @H©O | root | ^ |
| LENA_USER_GROUP | ¥ Manager %» 5 ` @~ OS ` @Hœ• | root | ^ |

| < = e f | ' (| . / g | e = h i |
|---|--|--------------------------|------------|
| LENA_HEALTH_CHECK | ¥ Health Check • € – É | N | ^ |
| LENA_HEALTH_CHECK_WAS_URL | ¥ Health Check @ J † s Og | /tie/lenaHealthCheck.jsp | ^ |
| LENA_HEALTH_CHECK_INITIAL_DELAY_MILLI SEC | ¥ LENA Agent %» † Ö Health Check ¾± ; 6%¾%, Server %» ¾%L dg * % Ä V | 60000 (milliseconds) | ^ |
| LENA_HEALTH_CHECK_TIMEOUT_MILLI SEC | ¥ Health Check - w Timeout | 5000 (milliseconds) | ^ |
| LENA_HEALTH_CHECK_FAILURE_THRESHOLD | ¥ Health Check ^ Q I © 4 | 5 | ^ |
| LENA_HEALTH_CHECK_TERM_EXECUTION | ¥ Health Check ^ Q I © 4 Ä S¾, ö Ð± ´ • € – É | true | ^ |
| LENA_HEALTH_CHECK_TERM_EXECUTION_SCRIPT | ¥ Health Check ^ Q I © 4 Ä S¾, ö Ð± ´ script Og | stop-container | ^ |
| LENA_HEALTH_CHECK_TERM_EXECUTION_INTERVAL | ¥ Health Check ^ Q I © 4 Ä S¾, ö Ð± ´ • € Î % | 300 (seconds) | ^ |

!

¥ LENA_CONFIG_TEMPLATE_ID: Revision t p \$ ° wr = * ¼, ° w¾ Default Revision 9(D\$ o 9.

¥ LENA_CONTRACT_CODE: License b x n < è 5 ` @GN, † ç † b x * s ½ L ^ ý y †, | 9(D\$ r • 1 o 9.

¥ JAVA_DOMAIN_CACHE_TTL: † ç † 30 G` L ¾
 \${JAVA_HOME}/jre/lib/security/java.security ØÙ: networkaddress.cache.ttl ç L Ì ^ . 9.

¥ LOG_OUTPUT_TYPE: Server 30 ØÙ 9(D\$ \ @¾ 9(D\$. 30 ØÙ: Log 30 † \ @ o 9.

Directory ! j

LENA Image %P %! 3 4 Ä 4 \$ Øusr/local/lena † N, œ * Ä m ¢ \$ 9 RS T 9.

| k l m n (\${LENA_HOME} op) | ' (|) * |
|-------------------------------|--------------------------------|-------|
| bin | Node Agent: Start/Stop scripts | î ` @ |
| depot | 3 4' Ä. Local Repository | î ` @ |
| etc | %P º P Og > 30 ØÙ | |

| | | |
|-----------------------------|---|-------|
| kl mn (\${LENA_HOME} op) | ' (|) * |
| license | License Og' hi * \$ a b c i | |
| logs | LENA hi @ Dœ ØÙ ÚÇ1 | |
| modules | LENA EF ÆÈ: ÚÇ1 Home | |
| clena-agent | Node Agent ^ €5 , - . ÆÈ † Ä4 * \$ ^ D | î ` @ |
| servers/webServer | Server 34 Home, \${LENA_SERVER_HOME} | |
| dbin | Server Start / Stop / hi @ ^ € Script ÚÇ1 | |
| dconf | Server 300g ÚÇ1 | |
| ddumps | Dump ØÙ ÚÇ1 | |
| dhook | Life-Cycle Hook Shell ØÙ ÚÇ1 | |
| dlib | Server ^ € Library ÚÇ1 | |
| dlogs | Log ØÙ ÚÇ1 | |
| dtemp | ‡ ´ @ I ¾ a b c i | |
| dwebapps | %! Application Deployment a b c i | |
| dwork | JSP Servlet Ĩ] 1 > yØÙ / S ÚÇ1 | |
| tmp | LENA hi @ I ¾ a b c i | |

Log & Dump q r

Log\$ Standard Out / ErrorD 5j * \$ {console} . S FileD 5j * \$ {file} . † s^ GN,] ^ Ĩ •
LOG_OUTPUT_TYPE0: Ç L {console} ' \$ {file}D 30 VQD# 5j . L ;] ~ • t 9.

1. Console 5j

Ù&\QD Container] ^ 5# h † Å@G\$. † 9. Server: Application Log, Access Log, GC Log' ÆW Standard OutQD 5j . 9. Docker5 30o Log Driver5 : 7 Node(Host): sOÄ4 (Docker: %! Log ØÙ Ä4 : /var/lib/docker/containers/[container-id]/[container-id]-json.log)5 ÚÇG- Í FluentD2 TA Log Aggregator5 : 7 • 1 / ÚÇ* – kl hi ' ~ • t 9.

2. ØÙ 5j

ØÙ 5j 30 ¾ LogØÙ > Dump ØÙA \${LENA_HOME}/servers/appServer/logs a b c i * Ä5 ØÙD ÚÇGN, u Log ØÙA logrotate305 : 7 Daily rolling o 9.
5j G\$ Log ØÙ: bcS 5j ØÙÜA Wv ~ 2 T 9.

| Log QR | qr ps |
|-----------------|-------------------------------------|
| Access Log | access_appServer_\${HOSTNAME}.log |
| GC Log | gc_appServer_\${HOSTNAME}.log |
| Application Log | appServer_lena-\${HOSTNAME}.out.log |

† LogØÙÜA Guest OS5 34o logrotate5 : 7 I Ù Rolling o 9.

ØÙ %& Log' 5j ~ ^ ý, BC: Log Aggregator' k 7 Log' • 1 / £ * \$ Loghi Stack (ELK, EFK Stack—)L mn * \$ ê † Ù&\ † 9. † ' Ä7# \$ Fluent-BitS T ASide-car Container' →r * –

Log' • 1 * \$ · ¸ † Û&\† 9.

DumpØÛA 9R Ä45 ° n o 9.

¥ Dump Home : \${LENA_HOME}/servers/appServer/dumps/ \${HOSTNAME}
\${HOSTNAME} a b c i \$ ÂÉ VolumeQD DumpØÛL ÚÇ~) ContainerB mÊL Ä. ê† 9.
Dump b c B ÚÇ Ä4\$ 9RS T 9.
¥ Heap Dump : \${DUMP_HOME}/hdump
¥ Thread Dump : \${DUMP_HOME}/tdump
¥ Service Dump : \${DUMP_HOME}/sdump

Health Check

Health Check: %! ?@A ! " # [Manager Health Check](#) ÉÊL M¢. 9.
Û&\QD WAS\$ Http Get· ¸ QD Health Check * sµ, LENA WAS: %! Health Check· ¸ A TCP
Port <è· ¸ QD 3OGf † 9. †\$ %! LENA Image5 Business Applicationr z ĚGs ½{ %
) " †N, Biz Application† z ĚG`L ^ý5\$ 7' Application: \õ. Http Get Health Check
3OL ´ " †ñ * %' ±N. 9. EF Kubernetes Manifest ØÛ %P %! 3OA 9RS T 9.

1. Readiness Check
- [TCPSocketAction : port 8180

[initialDelaySeconds : xx (Application Pn5 xë gO , -)

[timeoutSeconds : xx (Application Pn5 xë gO , -)
2. Liveness Check
- [TCPSocketAction : port 8180

[initialDelaySeconds : xx (Application Pn5 xë gO , -)

[periodSeconds : xx (Application Pn5 xë gO , -)

!

¥ LENA WAS: ^ý Server: O' %» ö5 Service Portr Listen ' " D Ĭ ^o 9.
¥ Health Check' Ä. Page\$ Checkr nF * • O' \QD #{ | r EFG\$
êQD i &*j GáD, #{ | : Back-end (K : Database)] s O' \es'
i &~ • t \$ Page' , O* – \@* %' ±N. 9.

Server Configuration t n

Container2 WASr %» G\$ õ¹ \$ 9RS T 9.
Û&\QDServer 3OA WASr %» G% ; 5 \@Gf ô * ¼, 3OL \@* \$ ¾° A 1) Base Image5
UV* - Í 2) Container%» ¾° 5 &) * \$ · 6 3) Application Artifact5 UV* \$ · ¸ † t 9.

| ' 6 ZX ST | ' (| LENA . i LM |
|----------------|---|----------------|
| Base Image5 UV | Base Image ° n¾5 3OOg' COPY* – Image5 UV | ^ |

| | | |
|--------------------------|--|------------------------------|
| ' 6 Z X S T | ' (| LENA . i L M |
| %» ¾° 5 &) | Container %» ¾, ÂÉ Repository5# 3O Og' COPY | ^ |
| Application Artifact5 UV | Spring Boot: ^ý2 T† Application Artifact5 Server3O Ogr %1Û Gf UVGf Application Artifact: xU2 V \@ | ^ WAS(Embedded) ' k 7 E F |

LENA5# \$ Manager' k 7 î i mn. Server 3OOg' Image Build¾° †Í , Container%» ¾° 5 &) ~ • t N , †' Ä7#\$ Application / WEB Server Container mn†; 5 Manager' 34* N, Server 3OOg' mn* – ô . 9.†' C, Û * • 9RS T9.



Figure 14. Service Cluster 3Ohi > Container 3O \@

LENA: Server Cluster %=L †@* – Server: 3OL î i mn* N, †' Image ' \$ %» ¾° 5 &) * – ô . 9.

Server 3O5 6. ' Š r †\$ \$ B C D E F G \$ () H I J K L M ¢ . 9.

Container Image Build

LENAr Base Image' EF* sµ Project / Nu` O} ' \$ Architecture ~ P5 : 7 Base Image' ' Ð ' \$ ¿ ~D Build7ô * \$ ^ýr ° ~ • t 9. Architecture : ` / OSO5# LENA Image: Ĩ ^9Ä' de* N, Image hi O} / %P • p † , - * 9.

9RA ¿ ~D Base Image' Build7ô * \$ ª e † 9.

¥ OSr LENA %! EF Image2 Z Û4

¥ LENA 34 Ä4: Ĩ ^ (%! : /usr/local/lena)

¥ %P \@ ¾| ¨: Architecture ~ PS LENA EF %! Image: Z Û4 «¹ r] ^3O • O* \$ êµQD\$ Zr =

Ä ^ý5\$ LENA%8sª L k 7 Base Image' Ê° n * – ô . 9.

Wv K¾2 T† &• Ĩ ^† , - . ^ý, LENA Image' ' Ð* – Base Image' ° n (Build). 9.

¥ Application Artifact: × 4
¥ , - Library 3 4
¥ JDK Ĩ ^ (%¶ Image5:) ú†] ^Ĩ • • O • P † * Û ^ý)
¥ ^ € Command Script • O
¥ %P \ @ ¾ | ¨: Architecture ~ PS LENA EF %! Image: ZÛ4 «¹ r] ^3O • OµQD
r =. ^ý
‘ Š ^ € r † § \$! " # Base Image ° n ÉÊL M¢. 9.

Application \]

Container h° S Server h° 5#: Application Artifact ×U5 6. · ¸ L ŒW N_* –ô . 9.
Project5# ×U õ¹ r /OG• , Kubernetes Deployment ManifestĦ ECS: TaskO: ' /Oo · ¸ 5
xy 3O* –ô . 9.
WAS h° 5# ApplicationL Deployment* \$ · ¸ 5\$ 9R Wr s r t 9.

| Server t u \] ST | ' (|
|----------------------------|--|
| %! Deployment a b c i 5 ×U | \${LENA_SERVER_HOME}/webapps5 WAR ' \$ Directory' +` |
| ÄB Application 3O5 xë ×U | Manager' k 7 ÄB Application 3O Application3O ù (DocBase) Ä45 WAR ' \$ Directory' +` |

Container h° 5#: Deployment · ¸ A 9R Wr s t 9.

| Container \] ST | ' (|
|--------------------------------|--|
| Î Container Image5 ` ; ×U · ¸ | Base Image5 Application Artifact' ÑÒ +` / UV |
| Init Container Å@ %» ¾° ×U · ¸ | Init Container5# ? É 5 UVo Artifact ' \$ ÂÉ ÚÇ1 (Volume)5 t \$ Artifact' %» ¾° 5 Î Container5 +` |

2.7. Server QRE * A 3% Ð Embedded WAS

2.7.1. \]

Embedded WAS\$ Container5 ×UG¼, 3O5 xy Container ¥ÖÖ (Kubernetes, ECS—)5 &Û
' \$ +• Ä: Instancer »¾5 ×Uo9.

- 1. Service ©m
Embedded WAS\$ &Û/+• Ä: ContainerD ×UGN, †' ÂÉÍ Front-End5 Service* %
Ä7#\$ + &5 L/B} ~L • €* \$ Service' ×4* \$ ê† Û&\e · ¸ †9. Kubernetes:
^ý5\$ 34o Node: ÞO PortD #{ | * \$ NodePort, ÂÉ L/B' Å@* \$ LoadBalancer, ? É
NO IP' sO* \$ ClusterIp: #{ | b c † EFGN t Q¼, ECS: ^ý5\$ ALB' sO* \$
· ¸ † t 9.` ; 5 ApplicationL f n. · ¸ QD Service~s ` ; /O† , - * 9.

2. Instance • (Replica)

&Û Service' * \$ + • Ä: Embedded WAS: • \$ É* 5 xy r ĩ \ t ĩ , Ã% %» 7ô ~ Instance Ä• ' ` ; 5 O: * – , x U 3 O 5 &) * – ô . 9.

3. Service Mapping

ECS: ^ ý 5 \$ Service 5 # Ñ Ò L / B' s O ~ • t s μ, Kubernetes: ^ ý 5 \$ Key-Value D O: o label L % P Q D Mapping o L O: * – Service 2 I ³ . 9. ; < ¾ | ¤ ' 5 # ù + * N, () 5 j i . Mapping % P L • p * – x U 3 O 5 &) * – ô . 9.

2.7.2. 3`

Embedded WAS' () * % Ä 7 # , - . ` æ A 9 R S T 9.

Memory

Embedded WAS: Heap Memory Size \$ å 1 512 Mbyte' , - D * N Image 5 \$ 9 R S T † % ! 3 O † G f t 9.

¥ Heap Memory : 1024 Mbyte

¥ Metaspace Memory : 128 Mbyte

† ' ĩ ^ * N H * • 9 R] ^ ĩ • ' 3 O * – ¢ O . 9.

¥ LENA_JVM_HEAP_SIZE

¥ LENA_JVM_METASPACE_SIZE

!

Ä] ^ ĩ • : ç A MByte & Ä † ¼ & § ¾ Y H + ĩ m ĩ c " : UV (K : 1024 m) c , Q D s O G f ô . 9. UV † Z Û 4 * • \ @ G s ½ \$ 9.

Disk

LENA Image % P Q D ` @ G \$ Image è % \$ B 700 Mbyte D † \$ O S + J D K + L E N A + L i b r a r y : [I Q D Image: ' Ä Layer @ \] s U V . @ \ † 9.

() 5 , - . - r \ e Disk \$ Log' Ø Û . , Q D Ú Ç ~) Log @ \ S Application 1 | Ø Û (Artifact): @ \ L N _ * – ü O . 9.

2.7.3. ' 6

abcd

1. " ñ # è Î 1

Embedded WAS: " ñ # è Î 1 \$ Þ B. E B † * 9. N / W' 5 # Manager 2 Session Server' \ s D * \$ & • ú k ç r = * C Ž x 4 7 ô . 9.

2. #{ | Uñ

Embedded WAS: #{ | Uñ \$ 9 R S T † N O G f t 9.

[HTTP #{ | Port : 8180

Ä Uñ \$ LENA Image 5 N O G f t N , Ä É # { | ĩ ^ L * N H * \$ ^ ý LENA Image ĩ ^ g 9 \$ Container: % ! ` ' 5 xy Port q e r 3 O L k 7 ĩ ^ ~ ê L ± N . 9.

< = e f

Embedded WAS Container 5 \ @ r = . Î -] ^ ĩ • \$ 9 R S T 9.

| < = e f | ' (| . / g | e = hi |
|------------------------------|---|--------------------------|-----------|
| LENA_SERVICE_PORT | ¥ WAS # { Port | 8180 | ^ |
| LENA_JVM_HEAP_SIZE | ¥ Heap Memory è % s O | 1024m | ^ |
| LENA_JVM_METASPACE_SIZE | ¥ Metaspace Memory è % | 128m | ^ |
| LENA_JVM_OPTIONS | ¥ ` @H O: JVM OPTION | | ^ |
| LENA_MANAGER_ADDRESS | ¥ Manager Î 1 ¥ c _ : IP / Domain Î 1 : # { Uñ | | ^ |
| LENA_MANAGER_MONITORING_PORT | ¥ Manager Œ • Ž • Port Og | 16100 | ^ |
| LENA_CONFIG_TEMPLATE_ID | ¥ 3 O Ø Û ID ¥ c _ : Service Cluster Ü | | ^ |
| LENA_SPRING_PROFILES_ACTIVE | ¥ SPRING PROFILE (PROFILE 3 O ý , • Ä r r Ç € R. & § ¾ † Ç L • O 7 ô V) | default | ^ |
| LOG_OUTPUT_TYPE | ¥ LOG 5 j b c ¥ X @ Ç : console ' \$ file | console | ^ |
| LENA_LOG_OUTPUT_DIR | ¥ Log Ø Û ° n Ä 4 | /usr/local/l ena/logs | ^ |
| LENA_SERVER_TYPE | ¥ Server b c | embedded | _ |
| LENA_HOME | ¥ LENA 3 4 Home ¥ Ç : /usr/local/l ena | (3 Ü M Ç) | _ |
| LENA_SERVICE_ENDPOINT | ¥ WASr Ð. # { : Î 1 | | ^ |
| LENA_APP_FILE | ¥ Application Jar Ø Û Ü | | ^ |
| LENA_APP_DIR | ¥ Application Jar a b c i Ü | /usr/local/l ena | ^ |
| LENA_EXCEPTION_ALERT_ENABLE | ¥ Exception ¯ ° ¾ Og • 1 – É | false | ^ |

| < = e f | ' (| . / g | e = hi |
|--|--|-------|-----------|
| LENA_EXCEPTION_CLASS_PATTERNS | ¥ • 1 6' Exception Class Og ' ; D – • Å Class ' .. / * – 3 O | | ^ |
| LENA_EXCEPTION_EXCLUDE_CLASS_PATTERNS | ¥ E Å 6' Exception Class Og ' ; D – • Å Class ' .. / * – 3 O | | ^ |
| LENA_FULLSTACK_HOOKED_EXCEPTION_ENABLE | ¥ Exception - ° ¾ Full Stack Trace • 1 – É | true | ^ |
| LENA_STUCKTHREAD_ALERT_ENABLE | ¥ Thread Stuck - ° ¾ Og • 1 – É | false | ^ |
| LENA_OOM_ALERT_ENABLE | ¥ Out Of Memory - ° ¾ Og • 1 – É | true | ^ |
| LENA_FULLGC_ALERT_ENABLE | ¥ Full GC - ° ¾ Og • 1 – É | false | ^ |
| LENA_REVERSE_TCP_CONNECTION_ENABLE | ¥ Reverse TCP Connection L k . Manager .. / ` @ – É | true | ^ |
| LENA_CONFIG_SERVER_URI | ¥ Spring Cloud Config %=L ` @* – Git Repository5# properties ØÙL r , f • t \$ Config Server URI Î 1 . ¥ † ç L Š G* • bootstrap.properties ØÙ5 URI Î 1 2 server.lena.config.enabled ç † trueD Š G, . | | ^ |

Directory ! j

LENA Image %P %! 3 4 Ä 4 \$ Øusr/local/lenaØ † N, œ * Ä m¢ \$ 9RS T 9.

| k l m n (\${LENA_HOME} op) | ' (|) * |
|-------------------------------|-----------------------|-----|
| logs | LENA hi @ Dœ ØÙ ÚÇ1 | |
| etc/info | Image Build Og ØÙ ÚÇ1 | |

Log & Dump q r

Log\$ Standard Out / ErrorD 5j * \$ ØconsoleØ . , S FileD 5j * \$ ØfileØ . , † sª GN,] ^ï •
ØLOG_OUTPUT_TYPEØ: ç L ØconsoleØ' \$ ØfileØD 3 O VQD# 5j . , L ;] ~ • t 9.

1. Console 5j

Ù&\QD Container] ^5# h† Å@G\$. , † 9. Server: Application Log, Access Log, GC
Log' ØEW Standard OutQD 5j . 9. Docker5 3 Oo Log Driver5 : 7 Node(Host): s OÄ4
(Docker: %! Log ØÙ Ä4 : /var/lib/docker/containers/[container-id]/[container-id]-json.log)5
ÚÇG- Í FluentD2 T A Log Aggregator5 : 7 • 1 / ÚÇ* – k l h i ' ~ • t 9.

2. ØÙ 5_i
ØÙ 5_i 3O ¾ LogØÙ > Dump ØÙA \${LENA_HOME}/servers/appServer/logs a b c i * Ä5
ØÙD ÚÇGN, u Log ØÙA logrotate3O5 : 7 Daily rolling o9.
5_i G\$ Log ØÙ: b c S 5_i ØÙÙA Wv ~ 2 T 9.

| Log QR | qr ps |
|-----------------|-------------------------------------|
| Access Log | access_appServer_\${HOSTNAME}.log |
| GC Log | gc_appServer_\${HOSTNAME}.log |
| Application Log | appServer_lena-\${HOSTNAME}.out.log |

† LogØÙÙA Guest OS5 34o logrotate5 : 7 I Ù Rolling o9.
ØÙ %& Log' 5_i ~ ^ý, BC: Log Aggregator' k 7 Log' • 1 / £ * \$ Loghi Stack (ELK, EFK Stack—)L mn * \$ ê† Ù&\†9. †' Ä7# \$ Fluent-BitS T ASide-car Container' -r * -
Log' • 1 * \$. † Ù&\†9.

DumpØÙA 9R Ä45 ° n o9.

¥ Dump Home : \${LENA_HOME}/servers/appServer/dumps/ \${HOSTNAME}
\${HOSTNAME} a b c i \$ ÄÉ VolumeQD DumpØÙL ÚÇ~) ContainerB mÊ L Ä. ê†9.
Dump b c B ÚÇ Ä4\$ 9RS T 9.

¥ Heap Dump : \${DUMP_HOME}/hdump
¥ Thread Dump : \${DUMP_HOME}/tdump
¥ Service Dump : \${DUMP_HOME}/sdump

Health Check

Health Check: %! ?@A ! " # [Manager Health Check](#) ÉÊ L M£. 9.
Ù&\QD WAS\$ Http Get· QD Health Check * sµ, LENA WAS: %! Health Check· A TCP
Port <è· QD 3OGf t9. †\$ %! LENA Image5 Business Applicationr z ÈGs ½{ %
) " †N, Biz Application† z ÈG` L ^ý5\$ 7' Application: \õ. Http Get Health Check
3OL ´ " †ñ * %' ±N. 9. EF Kubernetes Manifest ØÙ %P %! 3OA 9RS T 9.

1. Readiness Check
- [TCPSocketAction : port 8180

[initialDelaySeconds : xx (Application Pn5 xë gO , -)

[timeoutSeconds : xx (Application Pn5 xë gO , -)
2. Liveness Check
- [TCPSocketAction : port 8180

[initialDelaySeconds : xx (Application Pn5 xë gO , -)

[periodSeconds : xx (Application Pn5 xë gO , -)

!

¥ LENA WAS: ^ý Server: O' %» ö5 Service Portr Listen ´ " Dİ ^o9.
¥ Health Check' Ä. Page\$ Checkr nF * • O' \QD #{ |r EFG\$
êQD i &*j GáD, #{ | : Back-end (K : Database)] s O' \es'
i &~ • t \$ Page' , O* - \@* %' ±N. 9.

Container Image Build

LENAr Base Image' EF * sµ Project / Nu` O} ' \$ Architecture ~ P5 : 7 Base Image'
' Ð ' \$ ¿ ~D Build7ô * \$ ^ýr ~° ~ • t 9. Architecture : ` /OSO5# LENA Image:
İ ^9Ä' de * N, Image hi O} / %P • p† , - * 9.

9RA ¿ ~D Base Image' Build7ô * \$ ª e† 9.

¥ OSr LENA %! EF Image2 ZÛ4

¥ LENA 34 Ä4: İ ^ (%! : /usr/local/lena)

¥ %P \ @ ¾ | ¢: Architecture ~ PS LENA EF %! Image: ZÛ4 « ¹ r] ^3O • O* \$
êµQD\$ Zr =

Ä ^ý5\$ LENA%8sª L k 7 Base Image' Ê° n * – ô . 9.

Wv K¾2 T† &• İ ^† , - . ^ý, LENA Image' ' Ð* – Base Image' ° n (Build). 9.

¥ Application Artifact: × 4

¥ , - Library 34

¥ JDK İ ^ (%¶ Image5:) ú†] ^İ • • O • P † * Û ^ý)

¥ ^ € Command Script • O

¥ %P \ @ ¾ | ¢: Architecture ~ PS LENA EF %! Image: ZÛ4 « ¹ r] ^3O • OµQD
r =. ^ý

' Š ^ € r † § \$! " # Base Image ° n ÉÊL M¢. 9.

Application \]

Container h° S Server h° 5#: Application Artifact ×U5 6. · ¸ L ŒW N_* – ô . 9.
Project5# ×U õ¹ r /OG• , Kubernetes Deployment ManifestĪ ECS: TaskO: ' /Oo · ¸ 5
xy 3O* – ô . 9.

WAS h° 5# ApplicationL Deployment* \$ · ¸ 5\$ 9R W r s r t 9.

| Server t u \] ST | ' (|
|----------------------------|--|
| %! Deployment a b c i 5 ×U | \${LENA_APP_FILE} 3O \${LENA_HOME}5 Jar ' + ` |
| ÄB Application 3O5 xë ×U | \${LENA_APP_FILE} S \${LENA_APP_DIR} 3O \${LENA_APP_DIR}5 Jar ' + ` |

Container h° 5#: Deployment · ¸ A 9R W r s t 9.

| Container \] ST | ' (|
|--------------------------------|--|
| Î Container Image5 ` ; ×U · ¸ | Base Image5 Application Artifact' ÑÒ + ` / UV |
| Init Container Å@ %» ¾° ×U · ¸ | Init Container5# ? É5 UVo Artifact ' \$ ÂÉ ÚÇ1 (Volume)5 t \$ Artifact' %» ¾° 5 Î Container5 + ` |

2.8. Server QRE * A 3% Ð Web Server

2.8.1. \]

! " # òWAS ×Uó É Ê: 3ÜL M¢. 9.

2.8.2. 3`

Web Server' () * % Ä 7# , - . ` æA 9RS T 9.

Memory

Web Server: Heap Memory Size\$ å 1 512Mbyte' , - D * N, Agent: Heap Memory Size\$ 64~256MB' , - D . 9.

Disk

LENA Image %PQD ` @G\$ Image è%\$ ß 900 MbyteD † \$ OS + JDK
LENA + Library: [I QD Image: ' Ä Layer @\] s UV. @\ † 9.

() 5 , - . -r \ e Disk\$ Log' ØÛ · , QD ÚÇ~) Log@\ S Web i 1 | ØÛ (Artifact):
@\ L N_ * - üO. 9.

2.8.3. ' 6

abcd

1. " ñ#è Î 1
Web Server: " ñ#è Î 1 \$ ÞB. EB† * 9. N/W' 5# Manager2 WAS ' \$ WAS: Service
Endpoint' \ s D * \$ &· ú k ¿ r = * Cž 3O7ô . 9.

2. #{ | Uñ
Web Server: #{ | Uñ\$ 9RS T† NOGf t 9.
[HTTP #{ | Port : 7180
[HTTPS #{ | Port : 7543

Ä Uñ\$ LENA Image5 NOGf t N, ÄÉ #{ | İ ^L * NH * \$ ^ý LENA Image İ ^g 9\$
Container: %! ` ' 5 x y Port qer 3OL k 7 İ ^~ êL ±N. 9.

<=ef

WEB Server Container5 \ @ r =. Î -] ^İ • \$ 9RS T 9.

| <=ef | ' (| . / g | e= hi |
|--------------------------|---|-------|----------|
| LENA_MANAGER_ADD RESS | ¥ Manager Î 1 ¥ c , : IP / DomainÎ 1 : #{ Uñ | | ^ |
| LENA_MANAGER_KEY | ¥ Manager Open API ÒÐ c s | | ^ |

| < = e f | ' (| . / g | e = hi |
|---|---|---|-----------|
| LENA_SERVICE_PORT | ¥ WAS # { Port | 7180 | ^ |
| LENA_CONFIG_TEMPL ATE_DOWNLOAD | ¥ Manager DÉ Ž 3 O Ø Û 9 (D \$ – É ¥ X @ Ç : Y ' \$ N | | ^ |
| LENA_CONFIG_TEMPL ATE_ID | ¥ 3 O Ø Û ID ¥ c _ : Service Cluster Ü:Revision t p | | ^ |
| LENA_LICENSE_DOWN LOAD_URL | ¥ License 9 (D \$ URL ¥ k j Ç : manager ' \$ Nu` gbLicense9 (D \$ URI | manager | ^ |
| LENA_CONTRACT_CO DE | ¥ License - S h (o © ß H \$ D v p Û o Ç I . | | ^ |
| LOG_OUTPUT_TYPE | ¥ LOG 5 j b c ¥ X @ Ç : console ' \$ file | console | ^ |
| LENA_LOG_OUTPUT_DI R | ¥ Log Ø Û ° n Ä 4 | /usr/local/l enaw/serv ers/webSe rver/logs | ^ |
| LENA_AGENT_RUN | ¥ Node Agent % » – É | Y | ^ |
| LENA_SERVER_TYPE | ¥ Server b c | web | – |
| LENA_HOME | ¥ LENA 3 4 Home ¥ Ç : /usr/local/lena | (3 Ü M Ç) | – |
| LENA_SERVER_HOME | ¥ LENA # O 3 4 Ä 4 [Ç : /usr/local/lenaw/servers/webServer | (3 Ü M Ç) | – |
| LENA_USER | ¥ Manager % » 5 ` @ ~ OS ` @ H © O | root | ^ |
| LENA_USER_GROUP | ¥ Manager % » 5 ` @ ~ OS ` @ H œ • | root | ^ |
| LENA_HEALTH_CHECK | ¥ Health Check • € – É | N | ^ |
| LENA_HEALTH_CHECK _FAILURE_THRESHOLD | ¥ Health Check ^ Q I © 4 | 5 | ^ |

| | | | |
|---|---|----------------|--------|
| < = e f | ' (| . / g | e = hi |
| LENA_HEALTH_CHECK_TERM_EXECUTION | ¥ Health Check ^ Q I ©4 ÃS¾, öÐ± ´ • € – É | true | ^ |
| LENA_HEALTH_CHECK_TERM_EXECUTION_SCRIPT | ¥ Health Check ^ Q I ©4 ÃS¾, öÐ± ´ script Og | stop-container | ^ |
| LENA_HEALTH_CHECK_TERM_EXECUTION_INTERVAL | ¥ Health Check ^ Q I ©4 ÃS¾, öÐ± ´ • € î % | 300 (seconds) | ^ |

!

¥ LENA_CONFIG_TEMPLATE_ID: Revision t p\$ ° wr = * ¼, ° w¾ Default Revision 9(D\$ o9.

¥ LENA_CONTRACT_CODE: License b x n < è 5 ` @GN, t ç t b x * s ½L ^ ý y t, | 9(D\$ r • 1 o9.

¥ LOG_OUTPUT_TYPE: Server 3 O Ø Û 9(D\$ \ @¾ 9(D\$. 3 O Ø Û: Log 3 O t \ @ o9.

¥ LENA_AGENT_RUN: Web Server\$ Agent r % » G f ô Manager 5 – ž > Æ • ž • t r = * 9.

Directory ! j

LENA Image %P %! 3 4 Ä 4 \$ Ø /usr/local/lenaw Ø t N, œ * Ä m ¢ \$ 9 R S T 9.

| | | |
|--|---|-------|
| k l m n (\$ {LENA_HOME} o p) | ' (|) * |
| bin | Node Agent: Start/Stop scripts | |
| depot | 3 4' Ä. Local Repository | î ` @ |
| etc | %P ² P Og > 3 O Ø Û | |
| license | License Og' hi * \$ a b c i | |
| logs | LENA hi @ Dœ Ø Û Ú Ç 1 | |
| modules d lena-agent d lena-web-pe | LENA E F Æ È: Ú Ç 1 Home Node Agent ^ € 5 , - . Æ È t Ä 4 * \$ ^ D WEB Server Library | |
| servers/webServer d bin d cache d conf d hook d ht docs d logs d temp | Server 3 4 Home, \$ {LENA_SERVER_HOME} Server Start / Stop / hi @ ^ € Script Ú Ç 1 Cache Og Ú Ç 1 Server 3 O Og Ú Ç 1 Life-Cycle Hook Shell Ø Û Ú Ç 1 %! Web i 1 Ú Ç a b c i Log Ø Û Ú Ç 1 ± ´ @ I ¾ a b c i | |
| tmp | LENA hi @ I ¾ a b c i | |

Log

Log\$ Standard Out / ErrorD 5j * \$ {console} . S FileD 5j * \$ {file} . t s^a GN,] ^| •
LOG_OUTPUT_TYPE: 5 L {console} ' \$ {file}D 3O VQD# 5j . L ;] ~ • t 9.

1. Console 5j
! " # 0Log & Dump 5j 0L M\$. 9.

2. ØÙ 5j
ØÙ 5j 3O ¾ LogØÙ > Dump ØÙA \${LENA_HOME}/servers/webServer/logs a b c i * Ä5
ØÙD ÚÇGN, u Log ØÙA logrotate3O5 : 7 Daily rolling o9.

5j G\$ Log ØÙ: b c S 5j ØÙÜA Wv ~ 2 T 9.

| Log QR | qr vw(| ' (|
|------------|----------------------|---|
| Error Log | error_webServer.log | Web # O < q Dœ |
| Access Log | access_webServer.log | Access Dœ |
| Trace Log | trace_webServer.log | # { ¬ \ @ Dœ, Containerc # O5\$ ` @* s ½R |
| NTrace Log | ntrace_webServer.log | |
| LSC Log | lsc_webServer.log | |

Health Check

Ù&\QD WAS\$ Http Get. QD Health Check * sµ, LENA Web Server: %! Health
Check. A TCP Port <è. QD 3OGf t 9. t \$ %! LENA Image5 Business Applicationr
z ÈGs ½{ %) " t N, Biz Application† z ÈG` L ^ý5\$ 7' Application: \ Õ. Http Get
Health Check 3OL ´ " t ñ * %' ±N. 9. EF Kubernetes Manifest ØÙ %P %! 3OA 9RS
T 9.

1. Readiness Check
[TCPSocketAction : port 7180
[initialDelaySeconds : xx (Application Þn5 xë gO , -)
[timeoutSeconds : xx (Application Þn5 xë gO , -)

2. Liveness Check
[TCPSocketAction : port 7180
[initialDelaySeconds : xx (Application Þn5 xë gO , -)
[periodSeconds : xx (Application Þn5 xë gO , -)

Server Configuration t n

! " # 0Server Configuration h i 0' M\$. 9.

Container Image Build

! " # 0Container Image Build0' M\$. 9.

Chapter 3. ' s , J 3%

3.1. ' s x)

¥ <Container ()] ^ ° L>

34' Ä7#\$ Kubernetes, ECS— Container (@â • t\$] ^† ý, mnGf tfô . 9. œi N, CLI] ^5# 7'] ^5 Ö...~ • t\$ \ö. ±. S Profile (K : Kubernetes Config) 3O† , - * ¼, Manager ' \$ Session Server' VM5 34~ ^ý 34 6' VM† P{ Gf tfô . 9.

¥ <Network] ^ ° L>

¾| α mn Server% k¿† r =. s ° L* -ô . 9. Container N/W ? É5 ÆW 34G\$ ^ý\$ Û&\QD k¿ Eß† *sµ, VM ãl . ¾ QD mnG\$ ^ý Manager Server, Web Server, WAS, Session Server% k¿† r =. s ` ; de† , - * 9. 9RA LENA Image %POD Service ' \$ ' þ k¿L Ä7 , - . N/W Port † 9.

| Source | Target | Port | XY |
|------------------------|----------------|------------------------------------|-------------------------------|
| WEB/WAS/Session Server | Manager | TCP 7700 TCP 16100 UDP 16100 | Manager #{ EF Æ• Ž• , —ž |
| Bastion, hi H PC | Manager | TCP 7700 | Manager #{ EF |
| L/B , Front End | WEB Server | TCP 7180 | #{ EF |
| L/B , Front End | WAS | TCP 8180 | #{ EF |
| WAS | Session Server | TCP 5180 | #{ EF |

¥ <Image Ö...] ^ ° L>

LENA FÄ Image' Å@*% Ä7#\$ Container ¥ÖÖ5# Docker Hub5 ÖÐ† r = 7ô . 9. µß Ö...† Zr = * 9• , Ö... r =.] ^5# LENA Image' Pull ÁN (docker pull), †' ØÜD ÛÇ* - (docker save) 7' ¥ÖÖ5# Ö... r =.] ^5 \Ë (docker load) * -ô . 9. Wv\$ †' Zi * \$ K¾† 9.

```
$ sudo docker pull l enacloud/lena-cluster: {TAG_NAME}
... <5j ° w>
$ sudo docker save -o lena-cluster.tar l enacloud/lena-cluster: {TAG_NAME}
... <5j ° w>
$ sudo docker load -i lena-cluster.tar
... <5j ° w>
```

3.2. Base Image y "

WAS Í WEB Server: ^ý Base Image' È° n7ô , - r ° ~ • t 9. 9R5#\$ LENA Image' ' Ð* - Base Image' ° n(Build) 5 67 3Ü. 9. Base Image° nA LENA Image' Å@* \$ DockerfileL ‡ n* N, † ØÜD Docker Image' ° n* N F b Repository5 —ž * \$; < SO† 9.

K' G\$ Base Image: Î - Ĩ ^ ` aA 9RS TA Caser t 9.

1. , • Library -r

OSB 34 ÜÝf (yum, apt-get —)' †@* - Í , ÑÒ 34. 9.

2. JDK: Ĩ ^

LENA Base Image: JDK' Ĩ ^* NH ~) , 34 ÜÝf ' \$ ÑÒ UE 7E · , QD 34. 9.

3. Application Deployment

Application Deployment%! Ä4\$ /usr/local/lena/server/appServer/webappsł † 9.

—%5 ÑÒ WAR ØÛ ' \$ Explodedo Directory' +` *— Deployment ~ • † 9. † ^ý5\$ WAR ØÛÛ ' \$ DirectoryÛ† Context PathD sOo9.

9ë · , QD\$ ÄB Application 3OL k7 sOo Ä4(DocBase)5 WAR ' \$ Directory' +` *\$ · , QDC Deploy ~ • † 9. † ^ý5\$ 3O5# sO. Context PathD Application† #{ | o9.

' Š 3OA BCD EFG\$ l() HI J Kl: Application 3O ÉÊ L Mç. 9.

4. ^ € Command Script: • O

LENA: %! Container ^ € Command\$ \${LENA_HOME}/docker-entrypoint.sh † 9. Ä%] ^İ • 5 xë] ^3O > Server 3OØÛ 9(D\$, License 9(D\$, Server: ^ € —† ^ € G\$ Shell ScriptD#, Project] ^5 RCž Ĩ ^† , - . ^ý †' • O*— \@. 9.

3.2.1. Base Image y" 89

Base Image: ° n õ¹ \$ 9RS T9.

1. Dockerfile ‡ n

2. Docker Image O\$

3. Docker Image —ž

Dockerfile z "

9R H\$ \$ Base Image' ° n* % Ä. Dockerfile K¾† ¼, CentOS %&: Application Server ' %PQD ‡ n* † 9.

Dockerfile ± ¥

```

# This is a sample of Dockerfile to build project's Base Image with LENA
template.
# LENA image provides just basic environment to run LENA WAS.
# The project and the customer are responsible for optimization or change the
environments.
# Before building it, you need to check the proper LENA image repository &
tag.
FROM docker.io/lenacloud/lena-cluster:{TAG_NAME}

# Change or add JDK & packages as your own policy.
# RUN yum update -y
# RUN yum install -y java-1.8.0-openjdk-devel.x86_64

# The service address of LENA manager.
ENV LENA_MANAGER_ADDRESS lena-manager.namespace.svc.cluster.local:7700

# The key to access LENA manager.
ENV LENA_MANAGER_KEY you_manager_key_from_manager_user_admin_menu

# Id of template. Format is <Service Container Name>:<Version>.
ENV LENA_CONFIG_TEMPLATE_ID was-cluster_01:1

# To download and validate your license, LENA_CONTRACT_CODE is required.
# You can acquired it via LENA supplier.
# ENV LENA_CONTRACT_CODE your_own_lena_contract_code

# Download & change template files from manager.
RUN ${LENA_HOME}/docker-entrypoint.sh download_template

# If you have your own license file, copy it.
#COPY license.xml ${LENA_HOME}/license/

# Or If you uploaded your own container license file to manager, you can
download it from manager
# Caution!. After downloading you need to validate the file with like
'xmllint' command.
# RUN ${LENA_HOME}/docker-entrypoint.sh download_license

# Copy your application source to deployment path.
# COPY application.war /usr/local/lena/servers/appServer/webapps/

```

Ä ± ¥ Dockerfile: m¢\$ 9RS T9.

Table 1. ± ¥ Dockerfile: a B 3Ü

| | |
|-------------------------|---|
| ! " { | ' (|
| ' Ä Layer Image s O | <pre><m" >FROM \${' Ä Image} ' Ä ImageD LENA Image ' \$ LENA Image' ' ÐÁA Image ^ Project5# , f. Base Image: Repository/Tag' k i . 9.</pre> |
| Library -r 34 | <pre><m" > RUN yum install \${-r Library} OS BD QXs 34 ÜÝf' Ä@* - , - Library' -r * N, QXs B BC: 34 · 6† , - ~ ^ý 7' QXs: r †§5 xy Library' -r 34. 9. <JDK Ê 34> JDKC Library -r 2 » Û. . , QD 34~ · t 9. 9µ, LENA: ServerÛ † JDK' MÇ * N t Qá D, 9R: 300gC Ĩ ^ * - ô . 9. * Symbolic Link /usr/lib/jvm/java0L 34o JDK Ä45 RCž Ê 30 * - ô . 9. Ä Og\$ 9R ÉÊ 5# ` @ * N t 9. ¥] ^ Ĩ • \${JAVA_HOME} ¥ JDK 34Ä 4 : \${LENA_HOME}/etc/info/ java-home.info Ç ¥ Server] ^ 3O : \${LENA_HOME}/env.sh: JAVA_HOME Ç ¥ docker-entrpoint.sh5# domain cacheÇ TTL 3O</pre> |
|] ^ Ĩ • 3O | <pre><m" > ENV \${key} \${value} * Ä: 300g ' \$ y † , ' 9(D\$ Á% Ä.] ^ Ĩ • ' 3O. 9. - %5 3Oo Ç A Container %» ¾° 5C ` @â • t 9. ¥ LENA_MANAGER_ADDRESS : 34o Manager: Service Î 1 ¥ LENA_MANAGER_KEY : Open APID Manager Ò...¾ , - . e%cs (Manager: Admin > Users ² J 5# der =) ¥ LENA_CONFIG_TEMPLATE_ID : 9(D\$ ~ 300g , BH, (Service Cluster † @ + 0:0 + Revision t p) ¥ ENV LENA_CONTRACT_CODE : y † , 9(D\$' Ä. ©ß H\$</pre> |
| 300g 9(D\$ | <pre><m" > RUN \${LENA_HOME}/docker-entrpoint.sh download_template ManagerD É Ž 300g' Download . 9. curl / wget —: ÜÝQD Open API' p5. 9. Manager9(D\$¾ ö, UEØÛ 7E, Mod/Owner/Group • OL ^ €. 9.</pre> |
| y † , + ` ' \$ 9(D\$ | <pre><m" > RUN curl -o \\${LENA_HOME}/license/license.xml É<† * ° w> ManagerDÉ Ž License' 9(D\$ Á\$9. 9(D\$ ö XML Validation > Owner/Group • OL . 9. COPY license.xml \${LENA_HOME}/license/ BCD ^ - ÄA Licenser t L ^ý Container ? ÉD + ` . 9.</pre> |

| | |
|----------------------------|---|
| ! " { | ' (|
| Application Artifact +` | <pre><m" > COPY application.war \${LENA_SERVER_HOME}/webapps/ Application Artifact (KE5#\$ application.war) ØÛL DeployÄ45 +` . 9. %! Ä4\$ Wv 2 T 9. ¥ WAS : \${LENA_SERVER_HOME}/webapps/ ¥ WEB Server : \${LENA_SERVER_HOME}/webapps/htdocs Ä Ä4\$ ÄB Server > Application 3O5 : 7 Ĩ ^r = * 9.</pre> |

Šno DockerfileL %&QD Image' Build~ • t Q¼ (docker build), Build ö Docker Hub Í ECRS
TA Registry5 —ž (docker register)* – Container Platform5# Ä@~ • t Cž . 9.

Dockerfile z " (w} ~6)

```
root ©O† W< Û&©OL ` @*Æ a * $ ^ý Base Image ° n¾ ñr \e ‡ ´ † , - * 9. Wv
‡ ¥A Centos7 L ` @* ¼ 'lena' Û&©OL ` @* $ 3† | 9. ©O° n / sudo 34• #D í €o9.
Container %» ¾ entrypoint script 5# root ±. ÜÝL • €~ • t Cž , ! SO5# sudo ±. L
É – * N, password 3O * † sudo ÜÝ† r = * Cž 3O. 9. entrypoint script r pl ä) sudo
` @¾ password k j * Cž Ĩ ^* – Û&©O: sudo ÜÝL E. . 9.
```

Dockerfile ‡ ¥

```
FROM docker.io/lenacloud/lena-cluster: {TAG_NAME}

# create account
ENV LENA_USER lena
RUN adduser ${LENA_USER}
RUN chown -R ${LENA_USER}:${LENA_USER} ${LENA_HOME}

# sudo auth setting
RUN yum install -y sudo && yum clean all
RUN usermod -aG wheel ${LENA_USER}
## nopasswd sudo on - off this option in docker entry point
RUN sed -i 's/# %wheel/%wheel/g' /etc/sudoers

USER ${LENA_USER}
```

Docker Image • €

9RA DockerfileD Image' Build* \$ ÜÝf † 9. ` ; 5 Image: Repository > Tag RuleL O: * – ö
* ¼, • sž eHç5 Dockerfile: Ä4' sO* \$ ê5 b: . 9. (* % K¾\$ Current Path5
Dockerfile† t L ^ý5 bx * 9.)

Docker Image OS ^ € ÜÝf (K¾)

```
$ docker build --no-cache --rm --tag [REPOSITORY[:TAG]] .
```

' %o ÜÝf 5 ` @o • < A 9RS T9. • < A Project] ^5 RCž , B* – ` @. 9.

¥ --no-cache: †; O\$5# ° no • ¾' ` @* s ½R

¥ --rm: †î s ° n5 nFÓL) I ¾ BC†D' ŸE

Docker Image • ,

9RA Image Registry5 Image' —ž * \$ ÜÝf †9. ` ; 5 Registry5 —ž r =. ±. L r í
` @H2 v p —] ^3OL , - D . 9.

Docker Image —ž ^ € ÜÝf (K¾)

```
$ docker push [OPTIONS] [REPOSITORY[:TAG]]
```

%P ' Š Docker hi ÜÝf \$ 9RF , Site' M¢. 9.

<https://docs.docker.com/engine/reference/commandline/docker/>

docker commitA ™Ě () ùe Docker Container' ImageD —ž * \$ ÜÝf †9.

() ùe Container' ImageD —ž (K¾)

```
$ docker commit [OPTIONS] CONTAINER [REPOSITORY[:TAG]]
```

Chapter 4. Kubernetes . } \]

4.1. \] , J 3%

4.1.1. \] ×)

LENA 34 ` ; 5 Kubernetes ^ €] ^† mnGf tfô . 9. 9R u a û† P{ Gfô ~ ` a†9.

1. LENA' m» ~ Kubernetes Cluster2 Namespace: mn
2. kubectlt† 34GN CLlr r =.] ^ • (
3. Kubernetes Cluster5 Ò...† r =* Cž Kubernetes 3O (~/.kube/config): mn
4. Ò... r =. Docker Repository (K : ECR, Docker Hub, ? É Docker Repository —)
5. (• <) Kubernetes Config ØÛ : LENA Manager5# Log > Terminal ÒÐL Ä7 Manager5 ´ D\$, -
6. LENA Container License > ©BHS

4.1.2. \] VW

Kubernetes\$ <3| ñ...†< CmBD 9æ. ×U . , † ¶Ë*sµ, ! " #5#\$ %! Cme
kubectIL Ä@* \$ êL %PQD 3Ü. 9.

Kubernetes\$ Resource: ×U . ,] ^3OOg, | ' —L %8. YAMlc , : ÜŠ ØÛL Ä@* –
×U/ÿE/' " †ñ' •€. 9. *%5 3ÜG\$ \${manifest-file.yaml}A † ÜŠ#' s' . 9. !
" #5#\$ ×U5 , - . å1. : ÜÝf > •< µL %8* áD, ' Š. ?@A FÀo Kubernetes
3Ü#' M€. 9.

zf namespaceU ' 6

*%: Æ8 ×Uh(‡´A namespace &ÄD †7fs áD, ×U' *% +# ‡´ 6' † G\$
namespace' 3O*–ô . 9.

```
$ kubectl config set-context --current --namespace=${namespace}
```

Ä: . , QD 3O*s ½L ¾ *%: Æ8 ^ € ÜÝf " 5 ð--namespace=\${namespace}ð m" L
–r *–ô . 9. 9RA æ K¾†9.

```
$ kubectl apply -f ${manifest-file.yaml} --namespace=${namespace}
```

Kubernetes Resource\] , f...†b

ðkubectl applyð ÜÝf ' k7 ×U' ^ €* N, æc , A 9RS T9.

```
$ kubectl apply -f ${manifest-file.yaml}
```

Ä ÜÝf \$ ×U àW• y, » Û namespace: » Û nameL r í Objectr ¶Ë ~ ^ý 7' Resource:

ÜŠ' ´ " † ñ . 9.

Kubernetes ResourceU ‡ +

ðkubectl deleteð ÜÝf D × Uo Resource' ÝE. 9.

```
$ kubectl delete ðf ${manifest-file.yaml}
```

Workload f...† b „ ^ %

Application, Web Server: ^ý Application Artifact: Ĩ ^ —5 : 7 × Uo Workload ´ " † ñ r , - ~ • † 9. Kubernetes5# \$ † ' Rolling· ¸ QD ´ " † ñ * \$ · 6L EF. 9.

9RA × Uo Workload' Rolling· ¸ QD ´ " † ñ * \$ ÜÝf † 9.

```
$ kubectl rollout restart ${workloadType}/${workloadName}
```

Ä ÜÝf' k 7 Workloadr " ¿ G¼ • 1 \e ÄO(Revision)† ° n o 9. Revision ž L ¢ E * \$ ÜÝf \$ 9RS T 9.

```
$ kubectl rollout history ${workloadType}/${workloadName}
```

9RA Ñ; ´ " D • e * \$ ÜÝf † 9.

```
$ kubectl rollout undo ${workloadType}/${workloadName}
```

9RA k j o RevisionQD • e * \$ ÜÝf † 9.

```
$ kubectl rollout undo ${workloadType}/${workloadName}
--to-revision=${revisionNo}
```

Ä ÜÝf 5# eHÇ \${workloadType}A Workload: b c QD æ Ç A (statefulset) (deployment) (daemonset) ù * Í † 9. \${workloadName}A Workload: † @† N, \${revisionNo}\$ ÄO(Revision) Ç † 9.

\] Š ResourceU < Æ

f Uo Workload, Service—: Kubernetes Object' de * \$ ÜÝf kubectl get † 9.

```
$ kubectl get ${resourceType}
```

Ä ÜÝf 5# \${resourceType}A (statefulsets) (deployments) (daemonsets) (services) (configMaps) 2 T A Kubernetes Resource b c ù * Í † 9.

4.2. Manager \]

4.2.1. ' 6 %&

. / ' 6 %&

Manager Container\$ 9RS TA ±N 3OL %PQD ×UGf ô . 9.

| ' 6 t • %& | ' 6 g / ' (|) * |
|----------------------|--|-----|
| Workload p q | StatefulSet | - |
| ReplicaÄ • | 1 | - |
| Container Port | TCP : 7700 UDP : 16100 TCP : 16100 | - |
| Volume Mount | a b c i /usr/local/lena/repository' ÄÉ Volume5 I ³ (Wv KE 5# \$ persistentVolumeClaim · ³ QD I ³) | - |
| Probe (Health Check) | HttpGetAction, /lena J t s p 5 | - |

ZXOu 76 %& Ð Workload t •

\@ ¾° Project] ^5 x y / O7# \@7ô ~ 3O - 12 † ö 3ÜG\$ Manifest ØÜ5 \@o
Sample ÇA 9RS T9.

| ' 6 t • %& | ' (| Sample g |
|-----------------|---|---|
| Container Image | ProjectB Architecture / O5 x y , Bo OS > JDK O; 5 R\$ LENA Manager Image | lenacloud/le na- manager:{TA G_NAME} |
| namespace | Kubernetes? > i \ œ• Ü | default |
| name | Workload: † @, † ÇA Pod † @ / Hostname: PrefixD ` @o9. | lena- manager |
| label | Service2: `` / , LM5 ` @G\$ LabelD Key: Value– QD O: o9. | type: sample- lena- manager |
| imagePullPolicy | † î s Pull O} QD 9R ù * Í ' , f. 9. ¥ Always : RepositoryDÉ Ž a' PullÁR ¥ IfNotPresent : Local5 Imager * L ^ý 5µ Pull ÁR ¥ Never : Pull Ás ½R | Always |
| %P] ^İ • |] ^İ • : 3OA ENV ' \$ Config MapQD 3O r = | (configMap · ³) |

30 r =.] ^İ • \$ 9RS T9.u] ^İ • 5 6. ' Š. 3ÜA ! " # Manager] ^İ • ÉÊL MΦ. 9.

| < = e f | ' (| Sample g |
|-----------------------------|--------------------------|---|
| LENA_JVM_HEAP_SIZE | Heap Memory è % s O | 1024m (%!) |
| LENA_JVM_METASPACE_SIZE | Metaspace Memory è % | 256m (%!) |
| LENA_MANAGER_DOMAIN_ENABLED | Domain Name Å n Ů – É | Y |
| LENA_MANAGER_ADDRESS | Service: Domain Î 1 : Uñ | lena-manager.default.svc.cluster.local:7700 |
| JAVA_DOMAIN_CACHE_TTL | Domain Î 1 Cache ¾ % (Ã) | 3 (%!) |

Z X Ou 7 6 %& Ð Servicet •

| ' 6 t • %& | ' (| Sample g |
|------------|---|--------------|
| namespace | Kubernetes? > i \ œ • Ü | default |
| name | Service' , B* \$ † @QD, namespace? 5# b Û. ç † f ô . 9. † ç A Service Domain Î 1 5 \ @ o 9. | lena-manager |
| type | ÃÉ D Service' 4 5 * \$. , QD, 9R ù * Í † 9. ¥ NodePort : k8sr 3 4 o Æ 8 Node: s O PortD #{ ¥ LoadBalancer : ÃÉ Loadbalancer' k. #{ ¥ ClusterIp : k8s Cluster? ; @QD, NO o IPD #{ | NodePort |

4.2.2. Manifest . } \]

Workload

Kubernetes5#: Container\$ Pod &ÄD 3 4 G¼, Kubernetes Object' %8. YAML ØÛc , : Manifest ØÛL ‡ n* – 3 4 * \$ è † Ü&\ e . , † 9.

9RA Manager' 3 4 * % Ä. Manifest ØÛ ‡ ¥ † N, ' Š ? @A + # 3 Ü o 3 O a : / O 5 x y Project] ^ 5 RCž İ ^ * – ` @ ~ • † 9.

LENA Manager WorkloadÜŠ (Manifest) ØÛ

```
---
---
api Version: apps/v1
kind: Stateful Set
metadata:
  É name: l ena-manager
```

```

spec:
  selector:
    matchLabels:
      type: l ena-manager
  serviceName: l ena-manager
  replicas: 1
  template:
    metadata:
      labels:
        type: l ena-manager
    spec:
      containers:
      - name: l ena-manager
        image: docker.io/l enacloud/l ena-manager: {TAG_NAME}
        imagePullPolicy: Always
        ports:
        - containerPort: 7700
        envFrom:
        - configMapRef:
            name: configmap-l ena-manager
        volumeMounts:
        - name: wsy-l ena-manager-repository
          mountPath: /usr/local/l ena/repository
        readinessProbe:
          httpGet:
            path: /l ena
            port: 7700
            initialDelaySeconds: 20
            timeoutSeconds: 1
          livenessProbe:
            httpGet:
              path: /l ena
              port: 7700
              initialDelaySeconds: 30
              periodSeconds: 5
        volumes:
        - name: wsy-l ena-manager-repository
          persistentVolumeClaim:
            claimName: l ena-manager-repository
        terminationGracePeriodSeconds: 0

---
apiVersion: v1
kind: ConfigMap
metadata:
  name: configmap-l ena-manager
data:
  LENA_MANAGER_DOMAIN_ENABLED: "Y"

```

```
Ê LENA_MANAGER_ADDRESS: "lena-manager.default.svc.cluster.local:7700"
```

¥ × U ^ €

× U \$ kubectl apply -f lena-manager-deployment-sample.yaml

```
$ kubectl apply -f lena-manager-deployment-sample.yaml
```

¥ × U / S de

× U o Workload \$ kubectl get -f lena-manager-deployment-sample.yaml

```
$ kubectl get statefulsets
```

| NAME | READY | AGE |
|--------------|-------|-----|
| lena-manager | 1/1 | 10s |

Service

9RA Manager Service' × U * % Ä. Manifest ØÙA 9RS T 9.

LENA Manager ServiceÜŠ (Manifest) ØÙ

```
---
apiVersion: v1
kind: Service
metadata:
  name: lena-manager
spec:
  selector:
    type: lena-manager
  ports:
    - name: manager-tcp
      port: 7700
      targetPort: 7700
      nodePort: 31848
      protocol: TCP
    - name: monitoring-tcp
      port: 16100
      targetPort: 16100
      protocol: TCP
    - name: monitoring-udp
      port: 16100
      targetPort: 16100
      protocol: UDP
  type: NodePort
```

¥ × U ^ €

×U\$ kubectl apply ÜÝQD ^ € . 9. ØÜÜ† lena-manager-service-sample.yaml† y • ×U ÜÝA 9RS T 9.

```
$ kubectl apply y ðf lena-manager-servi ce-sampl e. yaml
```

¥ ×U/S de
×Uo Workload\$ kubectl get ÜÝf ^ €L k 7 de ~ • t 9.

```
$ kubectl get services
NNAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
...
lena-manager NodePort 10.43.xx.xx <none> 7700: 30770/TCP, 16100: 31610/UDP
10s
```

4.2.3. Manager Ž N

Service bcL NodePortD 3O* † 9 • http://[Node IP]: [Node Port] / 5 ÒÐ* – Manager5 ÒÐ. 9.

4.3. Session Server \]

4.3.1. \] ×)

×U ; 5 Manager5 Session Server' —ž * % Ä 7 # \$ Manager5# ðService Clusterð' —ž * – Õ . 9. Manager: ðCluster > Session Clusterð² J Ä 4 5 # Session Server bc : ¿ ~ Service Cluster' ° n . 9.

Service Cluster° n > hi h (' Š . ?@A B C E F " # e ð () H I J K ðL M ¢ . 9.

4.3.2. ' 6 %&

. / ' 6 %&

Session Server Container\$ 9RS TA ±N 3OL %PQD ×UGf ô . 9.

| ' 6 t • %& | ' 6 g / ' (|) * |
|----------------|--|-----|
| Workload pq | StatefulSet | - |
| ReplicaÄ • | 2 (Primary, Secondary Server 2Ä) | - |
| Container Port | TCP : 5180 | - |
| Probe | ExecAction, \${LENA_SERVER_HOME}/health.sh p 5 | - |

ZXOu 7 6 %& Ð Workload t •

\@ ¾° Project] ^ 5 xy / O 7 # \@ 7 ô ~ 3 O - 1 2 † ö 3ÜG\$ Manifest ØÜ5 \@o Sample ç A 9RS T 9.

| ' 6 t • %& | ' (| Sample g |
|-----------------|---|-----------------------------------|
| Container Image | ProjectB Architecture / O5 x y , B o OS > JDK O; 5 R\$ LENA Manager Image | lenacloud/lena-session:{TAG_NAME} |
| namespace | Kubernetes? > i \ œ• Ü | default |
| name | Workload: † @, † ç A Pod † @ / Hostname: PrefixD ` @ o 9. | lena- session |
| label | Service2: `` / , L M5 ` @G\$ LabelD Key: Value– QD O: o 9. | type: lena-session |
| imagePullPolicy | † î s Pull O} | Always |
| %P] ^İ • |] ^İ • : 3OA ENV ' \$ Config MapQD 3O r = | (configMap · ,) |

3O r =.] ^İ • \$ 9RS T 9. u] ^İ • 5 6. ' Š. 3ÜA ! " # [Session \] ^İ • ÉÊL MÇ. 9.](#)

| < = e f | ' (| Sample g |
|---------------------------|---|--|
| LENA_JVM_HEAP_SIZE | ¥ Heap Memory è% s O | 1024m (%!) |
| LENA_CONFIG_TEMPLATE_ID | ¥ Service Cluster Ü : Revision No | SESSION-001 |
| LENA_MANAGER_ADDRESS | ¥ Service: Domain Î 1 : Uñ | sample-lena-manager.default.svc.cluster.local:7700 |
| JAVA_DOMAIN_CACHE_TTL | ¥ Domain Î 1 Cache ¾% (Ã) | 0 (%!) |
| LENA_SESSION_0_ADDRESS | ¥ Primary Session # O Î 1 , StatefulSet3OS Ù4Gf ô V | lena-session-0.default.svc.cluster.local |
| LENA_SESSION_1_ADDRESS | ¥ Secondary Session # O Î 1 , StatefulSet3OS Ù4Gf ô V | lena-session-1.default.svc.cluster.local |
| LENA_SESSION_EXPIRE_SEC | ¥ Session µl ¾% (Ã) | 1800 (%!) |
| LENA_CONFIG_SHARE_SESSION | ¥ Application % Session F b– É | N |

Z X Ou 76 %& Ð Servicet •

| ' 6 t • %& | ' (| Sample g |
|------------|------------------------|----------|
| namespace | Kubernetes? > i \ œ• Ü | default |

| ' 6 t • %& | ' (| Sample g |
|------------|--|--------------|
| name | Service' B* \$ †@QD, namespace? 5# bÛ. ç † f ô . 9. † ç A Service DomainÎ 1 5 \ @o9. | lena-session |
| type | ÂÉ D Service' 4 5* \$ · QD, Session Server5\$ s O* s ½\$9. | |

4.3.3. Manifest . } \]

Workload

Kubernetes5#: Container\$ Pod &ÄD 34G¼, Kubernetes Object' %8. YAML ØÛc , : Manifest ØÛL ‡ n* – 34* \$ è† Û&\e · , †9.

9RA Manager' 34* % Ä. Manifest ØÛ ‡ ¥† N, ' Š ?@A + # 3Ûo 3O a : /O5 x y Project] ^5 RCž Ĩ ^* – ` @~ • t9.

LENA Session WorkloadÛŠ (Manifest) ØÛ K¾

```

---
api Version: apps/v1
kind: Stateful Set
metadata:
  Ê name: l ena-sessi on
spec:
  Ê selector:
  Ê   matchLabel s:
  Ê     type: l ena-sessi on
  Ê serviceName: l ena-sessi on
  Ê repl i cas: 2
  Ê templ ate:
  Ê   metadata:
  Ê     label s:
  Ê       type: l ena-sessi on
  Ê   spec:
  Ê     contai ners:
  Ê     - name: l ena-sessi on
  Ê       image: docker. i o/ l enacloud/ l ena-sessi on: {TAG_NAME}
  Ê       imagePul l Pol i cy: Al ways
  Ê       ports:
  Ê       - contai nerPort: 5180
  Ê       envFrom:
  Ê       - confi gMapRef:
  Ê         name: confi gmap-l ena-sessi on
  Ê       tty: true
  Ê       readi nessProbe:
  Ê       exec:
  Ê       command:
  Ê       - /usr/ l ocal / l ena/servers/sessi onServer/heal th. sh

```

```

    Ê      i n i t i a l D e l a y S e c o n d s : 20
    Ê      p e r i o d S e c o n d s : 5
    Ê      l i v e n e s s P r o b e :
    Ê      e x e c :
    Ê      c o m m a n d :
    Ê      - / u s r / l o c a l / l e n a / s e r v e r s / s e s s i o n S e r v e r / h e a l t h . s h
    Ê      i n i t i a l D e l a y S e c o n d s : 30
    Ê      p e r i o d S e c o n d s : 5
    Ê      t e r m i n a t i o n G r a c e P e r i o d S e c o n d s : 0
    ---
    a p i V e r s i o n : v1
    k i n d : C o n f i g M a p
    m e t a d a t a :
    Ê n a m e : c o n f i g m a p - l e n a - s e s s i o n
    d a t a :
    Ê L E N A _ S E S S I O N _ 0 _ A D D R E S S : " l e n a - s e s s i o n - 0 . l e n a -
    s e s s i o n . d e f a u l t . s v c . c l u s t e r . l o c a l : 5180 "
    Ê L E N A _ S E S S I O N _ 1 _ A D D R E S S : " l e n a - s e s s i o n - 1 . l e n a -
    s e s s i o n . d e f a u l t . s v c . c l u s t e r . l o c a l : 5180 "
    Ê L E N A _ M A N A G E R _ A D D R E S S : " l e n a - m a n a g e r . d e f a u l t . s v c . c l u s t e r . l o c a l : 7700 "
    Ê L E N A _ S E S S I O N _ E X P I R E _ S E C : "1800"
    Ê L E N A _ C O N F I G _ T E M P L A T E _ I D : " S E S S I O N - 001 "
    Ê L E N A _ C O N F I G _ S H A R E _ S E S S I O N : " N "

```

¥ × U ^ €

× U \$ k u b e c t l a p p l y -f l e n a - s e s s i o n - d e p l o y m e n t - s a m p l e . y a m l

```
$ kubectl apply -f lena-session-deployment-sample.yaml
```

¥ × U / S d e >

× U o W o r k l o a d \$ k u b e c t l g e t -f l e n a - s e s s i o n - d e p l o y m e n t - s a m p l e . y a m l

```

$ kubectl get statefulsets
NAME READY AGE
lena-manager 1/1 30m
lena-session 2/2 10s

```

Service

9 R A S e s s i o n S e r v i c e ' × U * % Ä . M a n i f e s t Ø Û A 9 R S T 9 .

LENA Session ServiceÜŠ (Manifest) ØÜ K¾*

```
---
apiVersion: v1
kind: Service
metadata:
  name: lena-session
spec:
  selector:
    type: lena-session
  clusterIP: None
```

¥ xU^ €
xU\$ kubectl apply ÜÝQD ^ €. 9. ØÜÜ† lena-session-service-sample.yaml† y • xU
ÜÝA 9RS T9.

```
$ kubectl apply -f lena-session-service-sample.yaml
```

¥ xU/S de
xUo Workload\$ kubectl get ÜÝf ^ €L k7 de ~ • t9.

```
$ kubectl get services
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
...
lena-session ClusterIP None <none> <none> 10s
...
```

4.3.4. Server • , < Œ

Manager5 ØÐ* – ØCluster > Service ClusterØ ² J 5# 7' Service Cluster' , f* – O'
—ž —É' de. 9. **ServerList** • • L —~ * • * &5 2Ä: Session Serverr ¢EG\$ êL de. 9.

! | Server\$ Manager: Scheduler5 : 7 —ž GáD, å 6 15Ã ö H» —ž o9.

4.4. WAS \]

4.4.1. \] x)

xU ; 5 Manager5 WAS Server' —ž * % Ä7#\$ Manager5# ØService ClusterØ —ž * – Ö
. 9. Manager: ØCluster > Server ClusterØ ² J Ä45# WAS(Enterprise/SE) bc: ¿ ~ Service
Cluster' ° n. 9.

Service Cluster' ° n. ö Overview ™5# Service Endpoint' 3O. 9. Kubernetes: ^ý
Øhttp://\${ServiceÜ}.\${namespaceÜ}.svc.cluster.localhost:\${Service Uñ}Ø c, † o9. † çA Web
Server: VirtualHost: Proxy3O5# Ä@o9.

° no Service Cluster: Template ™5# WAS: 3OL • €. 9. 300g ÚÇ ö Template5 6.
RevisionL ° n. 9. Service Cluster° n > hi h(' Š. ?@A BCEF " #e Ø() H I J KØL

MÇ. 9.

4.4.2. ' 6 %&

. / ' 6 %&

WAS Container\$ 9RS TA ±N 3OL %PQD ×U o9..

| ' 6 t • %& | ' 6 g / ' (|) * |
|----------------|-------------|-----|
| Workload pq | Deployment | |
| Container Port | TCP : 8180 | |

ZXOu 76 %& Ð Workload t •

\@ ¾° Project] ^5 x y / O7# \@7ô ~ 3O - 12 †ö 3ÜG\$ Manifest ØÛ5 \@o Sample ÇA 9RS T9.

| ' 6 t • %& | ' (| Sample g |
|------------------|--|-----------------------------------|
| Container Image | ProjectB Architecture / O5 x y , Bo OS > JDK O; 5 R\$ LENA Manager Image | lenacloud/lena-cluster:{TAG_NAME} |
| namespace | Kubernetes? > i \ œ• Ü | default |
| name | Workload: †@, † ÇA Pod †@ / Hostname: PrefixD ` @o9. | lena-was |
| label | Service2: `` / , LM5 ` @G\$ LabelD Key: Value– QD O: o9. | type: lena-was |
| strategy:type | Deployment Update O} , RollingUpdate, RecreateÛ , f. UpdateO} ÜL %8 | RollingUpdate |
| imagePullPolic y | † î s Pull O} | Always |
| replica Å• | Container (Pod) Å• | 2 |
| Probe | HttpGetAction, <è Page, ¾± Delay ¾%, Î %\$ Application Pn5 RCŽ 3O , - | 00 p 5 |
| %P] ^İ • |] ^İ • : 3OA ENV ' \$ Config MapQD 3O r = | (configMap • ,) |

3O r =.] ^İ • \$ 9RS T9. u] ^İ • 5 6. ' Š. 3ÜA ! " # WAS] ^İ • ÉÊL MÇ. 9.

| < = e f | ' (| Sample g |
|-------------------------|-----------------------------------|-----------|
| LENA_CONFIG_TEMPLATE_ID | ¥ Service Cluster Ü : Revision No | WAS-001:1 |

| < = e f | ' (| Sample g |
|-----------------------------------|---|---|
| LENA_MANAGER_ADDRESS | ¥ Service: Domain Î 1 : Uñ (+ # 34o Manager: Service Î 1) | lena- manager.defau lt.svc.cluster.lo cal:7700 |
| LENA_MANAGER_KEY | ¥ LENA_MANAGER_KEY : Open APID Manager Ö...¾, - . e%CS ¥ Manager: Admin > Users ² J 5# de r = | (ÀB Manager5# de k i , -) |
| LENA_CONFIG_TEMPLATE_DO WNLOAD | ¥ 3O Og 9(D\$ – É | Y |
| LENA_CONTRACT_CODE | ¥ y t , 9(D\$' Ä. ©B H\$ ¥ y t , - - ¾ EFo H\$ ç | (ÀB H\$ de , -) |
| LENA_LICENSE_DOWNLOAD_U RL | | manager |
| JAVA_DOMAIN_CACHE_TTL | ¥ Domain Î 1 Cache ¾% (Ä) | 3 (%!) |

ZXOu 76 %& Ð Servicet •

| ' 6 t • %& | ' (| Sample g |
|------------|--|----------|
| namespace | Kubernetes? > i \ œ• Ü | default |
| name | Service' , B* \$ †@QD, namespace? 5# bÜ. ç † f ô . 9. † çA Service DomainÎ 1 5 \ @o9. | lena-was |
| type | ÂÉD Service' 45* \$. , ÂÉD 45~ - š † * Q• BC 3O† Z, - | |

4.4.3. Manifest . } \]

Workload

Kubernetes5#: Container\$ Pod &ÄD 34G¼, Kubernetes Object' %8. YAML ØÛc , :
Manifest ØÛL ‡n* – 34* \$ ê† Û&\e . , †9.

9RA Manager' 34* % Ä. Manifest ØÛ ‡¥†N, ' Š ?@A + # 3Üo 3O a : /O5
x y Project] ^5 RCŽ Ĩ ^* – ` @~ • t9.

LENA Session WorkloadÛŠ (Manifest) ØÛ K¾*

```
---
api Versi on: apps/v1
ki nd: Depl oyment
metadata:
  Ê name: l ena-was
spec:
```

```

  selector:
    matchLabels:
      type: l ena-was
  replicas: 2
  strategy:
    type: RollingUpdate
  minReadySeconds: 10
  revisionHistoryLimit: 1
  template:
    metadata:
      labels:
        type: l ena-was
    spec:
      containers:
      - name: l ena-was
        image: docker.io/l enacl oud/l ena-cl uster: {TAG_NAME}
        imagePullPolicy: Always
        ports:
        - containerPort: 8180
        envFrom:
        - configMapRef:
            name: configmap-l ena-was
        readinessProbe:
          httpGet:
            path: /
            port: 8180
          initialDelaySeconds: 10
          periodSeconds: 5
        livenessProbe:
          httpGet:
            path: /
            port: 8180
          initialDelaySeconds: 15
          periodSeconds: 5
      volumes:
      terminationGracePeriodSeconds: 0

---
apiVersion: v1
kind: ConfigMap
metadata:
  name: configmap-l ena-was
data:
  LENA_CONFIG_TEMPLATE_DOWNLOAD: "Y"
  LENA_CONFIG_TEMPLATE_ID: "WAS-001"
  LENA_MANAGER_ADDRESS: "l ena-manager.default.svc.cluster.local: 7700"
  LENA_MANAGER_KEY:
    "aSw7RMPSw15LeN%2FMZnrxzjgV0BzZe18i VHZbJ8CkdLI ea2Ecd8AI eK9oPCLXuW%3D%3D"

```

```

Ê LENA_LICENSE_DOWNLOAD_URL: "manager"
Ê LENA_CONTRACT_CODE: "pghzJJqTdzaGtTuASr8yfw=="
JAVA_DOMAIN_CACHE_TTL: "3"

```

¥ × U ^ € >

× U \$ kubectl apply ÜÝQD ^ €. 9. ØÜÜ† lena-was-deployment-sample.yaml† y • × U ÜÝA 9RS T 9.

```
$ kubectl apply -f lena-was-deployment-sample.yaml
```

¥ × U / S de >

× U o Workload \$ kubectl get ÜÝf ^ € L k 7 de ~ • t 9.

```

$ kubectl get deployments
NAME READY AGE
lena-was 2/2 10s

```

Service

9RA Application Service' × U * % Ä. Manifest ØÜA 9RS T 9.

LENA Session ServiceÜŠ (Manifest) ØÜ K ¾

```

---
apiVersion: v1
kind: Service
metadata:
  Ê name: lena-was
spec:
  Ê selector:
  Ê   type: lena-was
ports:
  Ê - port: 8180
  Ê   targetPort: 8180

```

¥ × U ^ €

× U \$ kubectl apply ÜÝQD ^ €. 9. ØÜÜ† lena-was-service-sample.yaml† y • × U ÜÝA 9RS T 9.

```
$ kubectl apply -f lena-was-service-sample.yaml
```

¥ × U / S de

× U o Workload \$ kubectl get ÜÝf ^ € L k 7 de ~ • t 9.

```
$ kubectl get services
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
...
lena-was ClusterIP 10.43.xx.xx <none> 8180/TCP 20s
```

4.4.4. Server • , < Ⓔ

Manager5 ÒÐ* – ðCluster > Service Clusterð ²J 5# 7' Service Cluster' , f* – O' –ž –É' de. 9. Server List • • L –~ * • * &5 2Ä: WASr ¢EG\$ êL de. 9.

! | Server\$ Manager: Scheduler5 : 7 –ž GáD, å 6 15Ã ö H» –ž o9.

4.5. Embedded WAS \]

4.5.1. \] ×)

×U ; 5 Manager5 Embedded WAS Server' –ž * % Ä7# \$ Manager5# ðService Clusterð' –ž * – Ö . 9. Manager: ðCluster > Server Clusterð ²J Ä45# WAS(Embedded) bc: ¿ ~ Service Cluster' ° n. 9.

4.5.2. ' 6 %&

. / ' 6 %&

WAS Container\$ 9RS TA ±N 3OL %PQD ×U o9..

| ' 6 t • %& | ' 6 g / ' (|) * |
|----------------|-------------|-----|
| Workload pq | Deployment | |
| Container Port | TCP : 8180 | |

ZXOu 76 %& Ð Workload t •

\@ ¾° Project] ^5 xy /O7# \@7ô ~ 3O - 12 †ö 3ÜG\$ Manifest ØÛ5 \@o Sample çA 9RS T9.

| ' 6 t • %& | ' (| Sample g |
|-----------------|---|------------------------------------|
| Container Image | ProjectB Architecture / O5 xy , Bo OS > JDK O; 5 R\$ LENA Manager Image | lenacloud/lena-embedded:{TAG_NAME} |
| namespace | Kubernetes? > i \ œ• Ü | default |
| name | Workload: †@, † çA Pod †@ / Hostname: PrefixD ` @o9. | lena-was |
| label | Service2: `` / , LM5 ` @G\$ LabelD Key: Value– QD O: o9. | type: lena-was |

| ' 6 t • %& | ' (| Sample g |
|------------------|--|------------------|
| strategy:type | Deployment Update O} , RollingUpdate, RecreateÜ , f. UpdateO} ÜL %8 | RollingUpdate |
| imagePullPolic y | † î s Pull O} | Always |
| replica Å • | Container (Pod) Å • | 2 |
| Probe | HttpGetAction, < è Page, ¾± Delay ¾%, Î %\$ Application Pn5 RCž 3O , - | 00p5 |
| %P] ^İ • |] ^İ • : 3OA ENV ' \$ Config MapQD 3O r = | (configMap • .) |

3O r =.] ^İ • \$ 9RS T9.u] ^İ • 5 6. ' Š. 3ÜA ! " # Embedded WAS] ^İ • ÉÊL MΦ. 9.

| < = e f | ' (| Sample g |
|-------------------------------|---|---|
| LENA_CONFIG_TEMPLATE_ID | ¥ Service Cluster Ü : Revision No | WAS-001:1 |
| LENA_MANAGER_ADDRESS | ¥ Service: Domain Î 1 : Uñ (+ # 34o Manager: Service Î 1) | lena-manager.defau lt.svc.cluster.lo cal:7700 |
| LENA_MANAGER_MONITORIN G_PORT | ¥ Manager Ć• Ž• Port Og | 16100 |
| LENA_APP_FILE | ¥ Application Jar ØÜ Ü | sample-app.jar |
| LENA_APP_DIR | ¥ Application Jar a b c i Ü | /usr/local/lena |

Z XOu 76 %& Ð Servicet •

| ' 6 t • %& | ' (| Sample g |
|------------|---|----------|
| namespace | Kubernetes? > i \ œ• Ü | default |
| name | Service' . B* \$ †@QD, namespace? 5# bÜ. ç † f ô . 9. † ç A Service DomainÎ 15 \ @o9. | lena-was |
| type | ÃÉD Service' 45* \$. , ÃÉD 45~ - š † * Q• BC 3O† Z, - | |

4.5.3. Manifest . } \]

Workload

Kubernetes5#: Container\$ Pod &ÄD 34G¼, Kubernetes Object' %8. YAML ØÜc , : Manifest ØÜL ‡ n* – 34* \$ ê † Ü&\e • . † 9.

9RA Manager' 34*% Ä. Manifest ØÛ ‡¥†N, ' Š ?@A +# 3Üo 3O a : /O5
x y Project] ^5 RCž Ĩ ^* – ` @~ • t 9.

LENA Embedded WAS WorkloadÜŠ (Manifest) ØÛ K¾*

```
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: lena-was
spec:
  selector:
    matchLabels:
      type: lena-was
  replicas: 2
  strategy:
    type: RollingUpdate
  minReadySeconds: 10
  revisionHistoryLimit: 1
  template:
    metadata:
      labels:
        type: lena-was
    spec:
      containers:
        - name: lena-was
          image: docker.io/lenacloud/lena-embedded: {TAG_NAME}
          imagePullPolicy: Always
          ports:
            - containerPort: 8180
          envFrom:
            - configMapRef:
                name: configmap-lena-was
          readinessProbe:
            httpGet:
              path: /
              port: 8180
            initialDelaySeconds: 10
            periodSeconds: 5
          livenessProbe:
            httpGet:
              path: /
              port: 8180
            initialDelaySeconds: 15
            periodSeconds: 5
      volumes:
        terminationGracePeriodSeconds: 0
---
```

```

apiVersion: v1
kind: ConfigMap
metadata:
  name: configmap-lena-was
data:
  LENA_CONFIG_TEMPLATE_ID: "WAS-001"
  LENA_MANAGER_ADDRESS: "lena-manager.default.svc.cluster.local:7700"
  LENA_APP_FILE: "sample-app.jar"

```

¥ × U ^ € >

× U \$ kubectl apply -f lena-was-deployment-sample.yaml -n lena-was

```
$ kubectl apply -f lena-was-deployment-sample.yaml
```

¥ × U / S de >

× U o Workload \$ kubectl get deployments -n lena-was

```

$ kubectl get deployments
NAME READY AGE
lena-was 2/2 10s

```

Service

9RA Application Service' × U * % Ä. Manifest ØÙA 9RS T 9.

LENA Session ServiceÜŠ (Manifest) ØÙ K ¾

```

---
apiVersion: v1
kind: Service
metadata:
  name: lena-was
spec:
  selector:
    type: lena-was
  ports:
    - port: 8180
      targetPort: 8180

```

¥ × U ^ €

× U \$ kubectl apply -f lena-was-service-sample.yaml -n lena-was

```
$ kubectl apply -f lena-was-service-sample.yaml
```



```
¥ × U / S de
× U o Workload$ kubectl get ÜÝf ^ € L k 7 de ~ • t 9.
```

```
$ kubectl get services
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
...
lena-was ClusterIP 10.43.xx.xx <none> 8180/TCP 20s
```

4.5.4. Server • , < Œ

Manager5 ÒÐ* – ðCluster > Service Clusterŀ ² J 5# 7' Service Cluster' , f* – O' –ž –É' de. 9. **Server List** • • L –~ * • * & 5 2À: WASr €EG\$ êL de. 9.

!

Server\$ Manager: Scheduler5 : 7 –ž G á D, á 6 15Ã ö H» –ž o 9.

4.6. Web Server \]

4.6.1. \] ×)

× U ; 5 Manager5 Web Server' –ž * % Ä 7# \$ Manager5# ðService Clusterŀ' –ž * – ô . 9. Manager: ðCluster > Server Clusterŀ ² J Ä 4 5# WEB server bc: ¿ ~ Service Cluster' ° n. 9.

° n o Service Cluster: Template ™ 5# Web Server3OL . 9. 3O ? @ ù + # ° n o WAS2 Web Server " ©' Ä 7# \$ Virtual Host ™ 5# Proxy 3O† , - * 9.

3OL ÚÇ öTemplate5 6. RevisionL ° n. 9. Revision† ° nGf ô ÚÇo 3O: Downloadr r = 7í 9.

Service Cluster° n > hi h(' Š. ?@A B C E F " # e ŀ() H I J K ŀL M € . 9.

4.6.2. ' 6 %&

. / ' 6 %&

Web Server Container\$ 9RS TA ±N 3OL %PQD ×UGf ô . 9.

| ' 6 t • %& | ' 6 g / ' (|) * |
|----------------|-------------|---------|
| Workload pq | Deployment | |
| Container Port | TCP : 5180 | ĩ ^ Z r |

ZXOu 76 %& Ð Workload t •

\@ ¾° Project] ^ 5 x y / O 7# \@ 7ô ~ 3O - 1 2 † ö 3ÜG\$ Manifest ØÛ5 \@o Sample ÇA 9RS T 9.

| ' 6 t • %& | ' (| Sample g |
|-----------------|--|-------------------------------|
| Container Image | ProjectB Architecture / O 5 x y , B o OS > JDK O; 5 R\$ LENA Manager Image | lenacloud/lena-web:{TAG_NAME} |

| | | |
|-------------------------|---|-----------------------------|
| ' 6 t • %& | ' (| Sample g |
| namespace | Kubernetes? > i \ œ• Ü | default |
| name | Workload: †@, † çA Pod †@ / Hostname: PrefixD ` @o9. | lena-web |
| label | Service2: `` / , LM5 ` @G\$ LabelD Key: Value– QD O: o9. | type: lena-web |
| strategy:type | Deployment Update O} | RollingUpdate |
| imagePullPolicy | † î s Pull O} | Always |
| replica Å• | Container (Pod) Å• | 2 |
| Probe (Health Check) | HttpGetAction · ¸ , < è Page, ¾ ‡ Delay¾%, Î %\$ Application Pn5 Rj 3O , - | 0 p5, 5Ã Î %, 15Ã/20Ã Delay |
| %P] ^İ • |] ^İ • : 3OA ENV ' \$ Config MapQD 3O r = | (configMap · ¸) |

3O r =.] ^İ • \$ 9RS T9. u] ^İ • 5 6. ' Š. 3ÜA ! " # [Web Server](#)] ^İ •
ÉÊL Mç. 9.

| | | |
|---------------------------------------|--|---|
| < = e f | ' (| Sample g |
| LENA_CONFIG_TE MPLATE_ID | ¥ Service Cluster Ü : Revision No | WEB-001:1 |
| LENA_MANAGER _ADDRESS | ¥ Service: Domain Î 1 : Uñ (+ # 34o Manager: Service Î 1) | lena- manager.default.s vc.cluster.local:77 00 |
| LENA_MANAGER _KEY | ¥ LENA_MANAGER_KEY : Open APID Manager Ö...¾ , - . e%oCS ¥ Manager: Admin > Users ² J 5# der = | (ÀB Manager5# de k j , -) |
| LENA_CONFIG_TE MPLATE_DOWNL OAD | ¥ 3O Og 9(D\$ – É | Y |
| LENA_CONTRACT _CODE | ¥ y †, 9(D\$' Ä. ©B H\$ ¥ y †, - - ¾ EFo H\$ ç | (ÀB H\$ de , -) |
| LENA_LICENSE_D OWNLOAD_URL | ¥ y †, 9(D\$ Ä4 | manager |
| LENA_RUN_AGEN T | ¥ Agent ^ €– É | Y |

Z XOu 76 %& Ð Servicet •

| ' 6 t • %& | ' (| Sample g |
|------------|--|--------------------------|
| namespace | Kubernetes? > i \ œ• Ü | default |
| name | Service' ı B* \$ †@QD, namespace? 5# bÛ. ç † f ô . 9. † ç A Service DomainÎ 1 5 \ @o9. | lena-web |
| type | ÃÉ D Service' 45* \$ · ı , ÃÉ D 45~ - š † * Q• BC 3O† Z, - | NodePort (port 31180) |

4.6.3. Manifest . } \]

Workload

Kubernetes5#: Container\$ Pod &ÄD 34G¼, Kubernetes Object' %8. YAML ØÛc ı : Manifest ØÛL ‡ n* – 34* \$ ê † Û&\e · ı † 9.

9RA Manager' 34* % Ä. Manifest ØÛ ‡ ¥† N, ' Š ?@A +# 3Üo 3O a : /O5 x y Project] ^5 RCž Ĩ ^* – ` @~ • t 9.

LENA Web WorkloadÛŠ (Manifest) ØÛ K¾

```

---
api Version: apps/v1
kind: Deployment
metadata:
  Ê name: lena-web
spec:
  Ê selector:
  Ê   matchLabels:
  Ê     type: lena-web
  Ê replicas: 1
  Ê strategy:
  Ê   type: RollingUpdate
  Ê minReadySeconds: 10
  Ê revisionHistoryLimit: 1
  Ê template:
  Ê   metadata:
  Ê     labels:
  Ê       type: lena-web
  Ê   spec:
  Ê     containers:
  Ê       - name: lena-web
  Ê         image: docker.io/lenacloud/lena-web: {TAG_NAME}
  Ê         imagePullPolicy: Always
  Ê         ports:
  Ê       - containerPort: 7180
  Ê         readinessProbe:
  Ê           httpGet:
  Ê             path: /

```

```

    port: 7180
    initialDelaySeconds: 5
    periodSeconds: 5
    livenessProbe:
      httpGet:
        path: /
        port: 7180
    initialDelaySeconds: 10
    periodSeconds: 10
    envFrom:
    - configMapRef:
        name: configmap-lena-web
    terminationGracePeriodSeconds: 0

---
apiVersion: v1
kind: ConfigMap
metadata:
  name: configmap-lena-web
data:
  LENA_MANAGER_ADDRESS: "lena-manager.default.svc.cluster.local:7700"
  LENA_AGENT_RUN: "Y"
  LENA_CONFIG_TEMPLATE_ID: "WEB-001:1"
  LENA_CONFIG_TEMPLATE_DOWNLOAD: "Y"
  LENA_MANAGER_KEY:
    "aSw7RMPSw15LeN%2FMZnrxzjgV0BzZe18iVHZbJ8CkdLIea2Ecd8AIeK9oPCLXuW%3D%3D"
  LENA_LICENSE_DOWNLOAD_URL: "manager"
  LENA_CONTRACT_CODE: "dX89RRxPk6/PBPqbUuYm7w=="

```

¥ × U ^ €
 × U \$ kubectl apply -f lena-web-deployment-sample.yaml † y • × U
 ÜÝA 9RS T 9.

```
$ kubectl apply -f lena-web-deployment-sample.yaml
```

¥ × U / S de
 × U o Workload\$ kubectl get -f lena-web-deployment-sample.yaml

```

$ kubectl get deployments
NAME READY AGE
lena-web 2/2 10m
lena-was 2/2 10s

```

Service

9RA Application Service' × U * % Ä. Manifest ØÙA 9RS T 9.

LENA Session ServiceÜŠ (Manifest) ØÜ K¾

```
---
apiVersion: v1
kind: Service
metadata:
  name: lena-web
spec:
  selector:
    type: lena-web
  ports:
    - nodePort: 31180
      port: 7180
      targetPort: 7180
      type: NodePort
```

¥ ×U^ €
 ×U\$ kubectl apply ÜÝQD ^ €. 9. ØÜÜ† lena-web-service-sample.yaml† y • ×U ÜÝA
 9RS T9.

```
$ kubectl apply -f lena-web-service-sample.yaml
```

¥ ×U/S de
 ×Uo Workload\$ kubectl get ÜÝf ^ €L k7 de ~ • t9.

```
$ kubectl get services
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
...
lena-web NodePort 10.43.xx.xx <none> 7180:31180/TCP 13h
...
```

4.6.4. Server • , < Æ

Manager5 ØÐ* – @Cluster > Service Cluster0 ²J5# 7' Service Cluster' , f* – O'
 –ž –É' de. 9. **Server List** • • L –~ * • * &5 2Ä: Web Serverr ¢EG\$ êL de. 9.

!

Server\$ Manager: Scheduler5 : 7 –ž GáD, å6 15Ã ö H» –ž o9

Chapter 5. ECS . } ' s

! Ç5#\$ ECS] ^5# LENA #O ' \$ LENA †î s %&: 2¥i 3†< L ContainerD ×U* \$. 6L 3Ü. 9. AWS ConsoleL k 7# 34* \$. 6L 3Ü. 9. ' . , Ü&\e ECS 3O5 6. ÉÊA EÂ* N, ECS] ^5# LENA' m» *% Ä7 , - . ?@µ 9> 9.

5.1. ECS '

ECS\$ AWS5# EF* \$ Container #{ | ¥ÖÖQD, Docker' †@* – EC2 e| ä| ' 5# Container' ×U/() * N, » Û Service' EF* \$ Container' Task &ÄD œf # Replica, Service Discovery, L/B, Auto-scale O} —L hi ~ • t \$ %=L EF. 9.

5.2. ' s x)

9RS TA ECS] ^††î i P{ Gf tf ô . 9.

¥ ECS Cluster > ECS ‡ ´ ±.

¥ ECS] ^5# Ö... r =. Docker Registry (K : ECR, Docker Hub)

–r \QD Container %& Manager' 34* % Ä7#\$ ÂÉ ÚÇ1r , - * N, †êA Manager Container: ° n/1î ¾5C %¶ " †Ž > ØÛL sÐ\QD bs* % Ä7 ` @o9.

¥ EFS — ÂÉ ÚÇ1 r =] ^ de

Container %& Manager > Session Server' 34* % Ä7#\$ Service Discovery %=† \@G\$] ^ (awsvpc %& EC2, Fargate > s^a Region)es ` ; de† , - * 9.

¥ ECS Service Discovery \@ r =] ^

5.3. Manager \]

ECS] ^5#C Manager Instance:) Ðn gÇL *% Ä7# 1) NOÎ 1 2) ÂÉ Volume W r s r , - * 9. NOÎ 1 \$ ECS: Service DiscoveryÍ ELB ¨ /L k 7# r =* ¼, ÂÉ VolumeA EFS ¨ /' k 7 EF r =* 9. * %5#\$ Service Discovery2 EFS' Å@. Manager ×U5 67 3Ü. 9.

5.3.1. ' 6 %&

. / ' 6 %&

Manager Container\$ 9RS TA ±N 3OL %PQD ×UGf ô . 9.

Table 2. ECS%& Manager 3 4 - ×U %P

| ' 6 t • %& | ' 6 g / ' (|) * |
|-------------------|--|-----|
| Service bc | Replica | - |
| ReplicaÂ • | 1 | - |
| Service Discovery | Service Discovery ` @ | - |
| Volume Mount | a b c i /usr/local/lena/repository' EFS5 ¨ / | - |

ZXOu 76 %&

\@ ¾° Project] ^5 xy /O7# \@7ô ~ 3O - 12 †ö 3ÜG\$ 34SO5# ` @G\$ Sample çA 9RS T9.

Table 3. ECS%& Manager 3 4 - \@¾° /O a

| ' 6 t • %& | ' (| Sample g |
|------------------------|--|-----------------------------------|
| Container Image | ProjectB Architecture / O5 xy , B o OS > JDK O; 5 R\$ LENA Manager Image | lenacloud/lena-manager:{TAG_NAME} |
| Probe (Health Check) | HttpGetAction, Ø/lenaØ J † s p5 | |
| Task > Service name | Task > Service : † @ | lena-manager |
| Service Namespace | ServiceL M¾ Domain Î 1 : SuffixD ` @ | local |
| Service Discovery Name | ServiceL M¾ Domain Î 1 : PrefixD ` @ | lena-manager |

3O r =.] ^İ • \$ 9RS T9. u] ^İ • 5 6. ' Š. 3ÜA ! " #: Manager] ^İ • ÉÊ Mç. 9.

Table 4. ECS%& Manager 3 4 -] ^İ •

| < = e f | ' (| Sample g |
|-----------------------------|----------------------------|-------------------------|
| LENA_JVM_HEAP_SIZE | ¥ Heap Memory è % s O | 1024m (%!) |
| LENA_JVM_METASPACE_SIZE | ¥ Metaspace Memoryè % | 256m (%!) |
| LENA_MANAGER_DOMAIN_ENABLED | ¥ Domain Name Å n Û – É | Y |
| LENA_MANAGER_ADDRESS | ¥ Service: Domain Î 1 : Uñ | lena-manager:local:7700 |
| JAVA_DOMAIN_CACHE_TTL | ¥ Domain Î 1 Cache ¾ % (Ã) | 3 (%!) |

5.3.2. Task ' 6

Task: † @, ±. —A Project: ~ P5 xy k j * Æ ± N * ¼, * %5# \$ Container O: ù LENA () L Ä7 , - . ÉÊµ 3Ü. 9. * %5 3ÜG\$ 3O: %PA 3O a ÉÊ: 3ÜL Mç. 9.

Task 6U

Container Og' UV* \$ Task' O: . 9.



Figure 15. ECS%& Manager 3 4 - Task O:

Volume ' h

Task O: 5# Manager Repository' ÚÇ~ EFS2 ÚÇÄ4' Og' ¬r . 9.

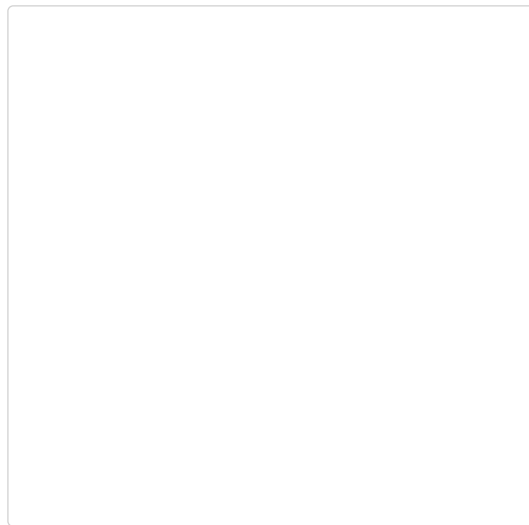


Figure 16. ECS%& Manager 3 4 - Volume ¬r

Container ' h

†@,†† s—: %! ContainerOg' kᵢ . 9.



Figure 17. ECS%& Manager 3 4 - Container ¬r

<=ef ' 6

Container 3Où] ^İ • ' ¬r . 9. Manager ?É5# Manager: Service Discovery Î 1' e , * s
• * áD †'] ^İ • LENA_MANAGER_ADDRESSED kᵢ L . 9.



Figure 18. ECS%& Manager 3 4 -] ^İ • 3O

Health Check ' 6

<http://localhost:7700/lena/> J † s' p5* \$ Health Check Og' k i . 9.

Volume " "

+ # ¬r . VolumeL Container? É Directory2 I ³ . 9.

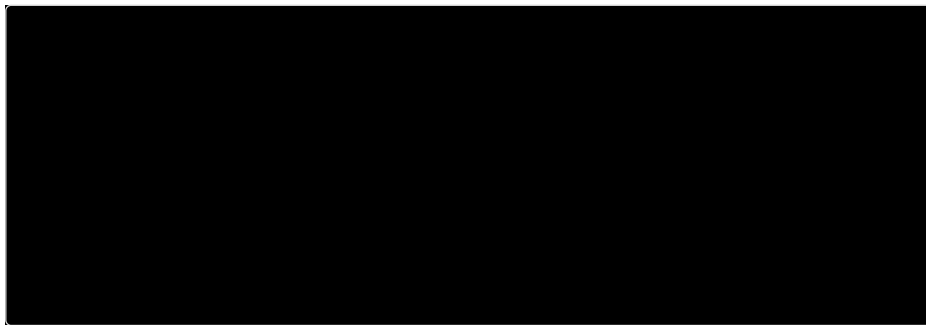


Figure 19. ECS%& Manager 3 4 - Volume I ³

5.3.3. Service ' 6

|) 1 6U

+ # O: . Task' ^ E %» /() * % Ä. Service' O: . 9.

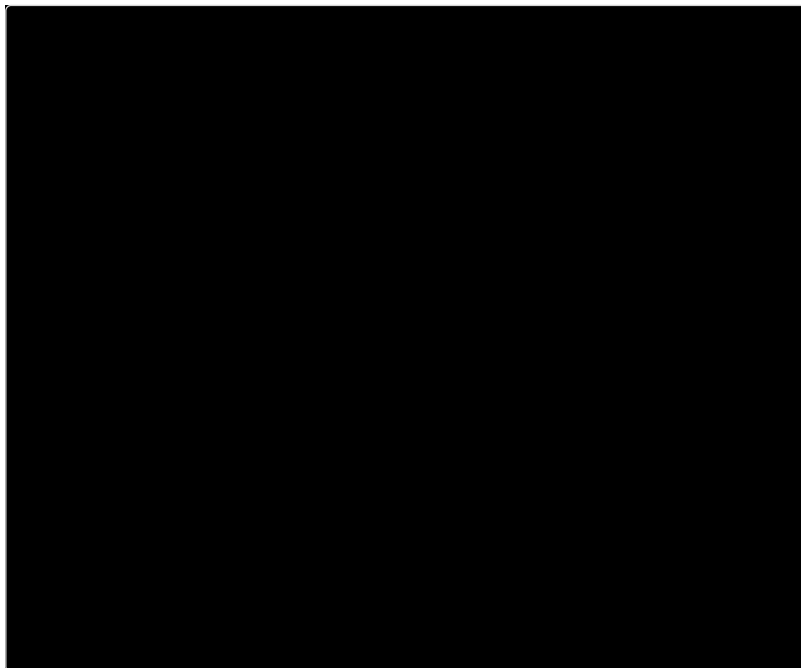


Figure 20. ECS%& Manager 3 4 - Service O:

|) 1 • – (Service Discovery) ' 6

AWS\$ ECS% # { | û L " / * – ` @ ~ • t C ž # { | L M (Service Discovery) %=L s^a . 9.
 Manager\$ Service Discovery %=L † @ * – N O o Î 1' dg* –, P Server: 3 O h i > Æ • Ž •
 %=L E F . 9.



Figure 21. ECS%& Manager 3 4 - Service Discovery 3 O

5.3.4. Service . — „ < Æ

Service %» A Service O: ' Ú Ç * • %» † ¾ † o 9. ECS Cluster: Û • 5 # # { | > ‡ ´ ™ 5 #
 () ù e Service 2 Task: ' " ' d e . 9.

Service ~ ™ < Æ

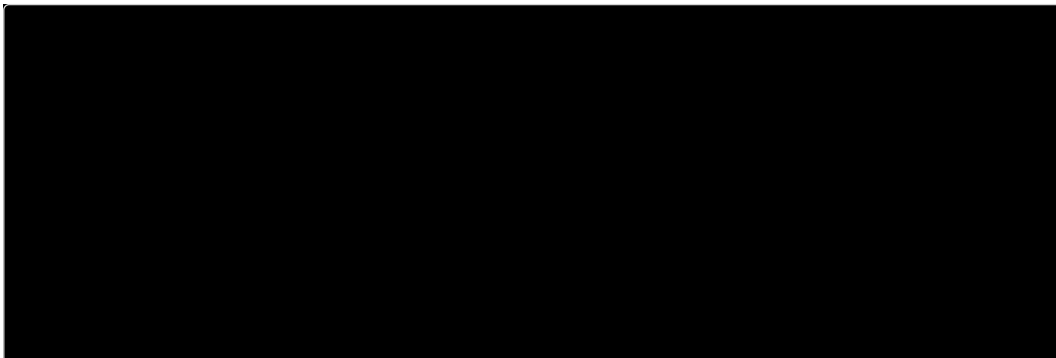


Figure 22. ECS%& Manager 3 4 - Service ' " d e

Task ~ ™ < Æ

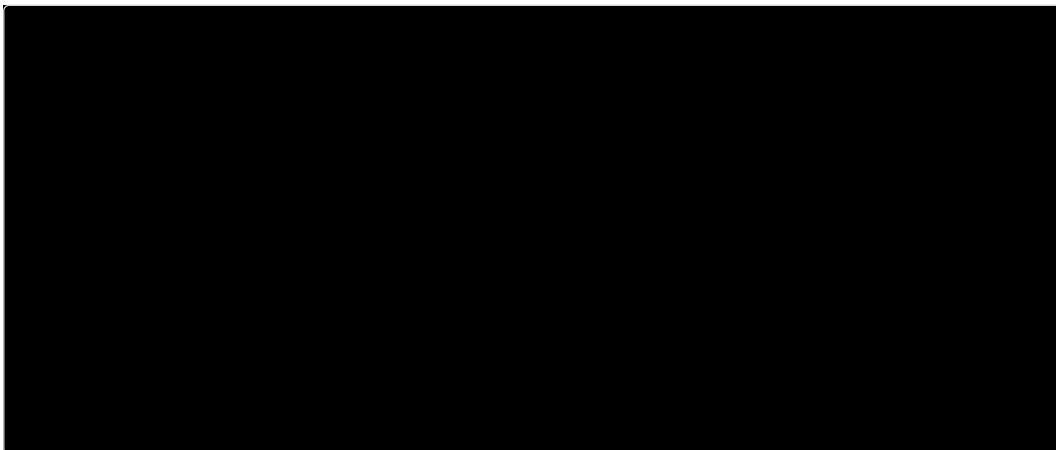


Figure 23. ECS%& Manager 3 4 - Task ' " d e

5.4. Session Server \]

ECS5#] ^5#C Session Server: # { | ' Ä 7 NOÎ 1r , - * 9. NO Î 1\$ ECS: Service DiscoveryÍ ELB ¨ / L k 7# r = * 9. 9R5#\$ Service Discovery ' Å@. Session Server xU5 6 7 3Ü. 9.

5.4.1. \] x)

! " # [Session Server xU P{](#) É Ê 3ÜL M¢. 9.

5.4.2. ' 6 %&

. / ' 6 %&

Session Server Container\$ 9RS TA ±N 3OL %PQD xUGf ô . 9.

Table 5. ECS %& Session Server xU %P

| ' 6 t • %& | ' 6 g / ' (|) * |
|----------------------|---------------------------------------|-----|
| Service p q | Replica | - |
| Service / ReplicaÅ • | Service 2Å, u Service B Replica 1Å | - |
| Container Port | TCP : 5180 | - |
| Probe | \${LENA_SERVER_HOME}/healt h.sh p5 | - |

ZXOu 7 6 %&

\ @ ¾° Project] ^5 x y / O7# \ @7ô ~ 3O - 1 2 † ö 3ÜG\$ Task > Service 3O5
\ @o Sample ç A 9RS T 9.

Table 6. ECS%& Session Server 3 4 - \ @¾° / O a

| ' 6 t • %& | ' (| Sample g |
|---------------------------|--|---------------------------------------|
| Container Image | ProjectB Architecture / O5 x y , B o OS > JDK O; 5 R\$ LENA Manager Image | lenacloud/lena- session:{TAG_NAME} |
| Task > Service name | Task > Service : † @ | lena-session |
| Service Namespace | ServiceL M¾ Domain Î 1 : SuffixD ` @ | local |
| Service Discovery Name | ServiceL M¾ Domain Î 1 : PrefixD ` @ | lena-session |

3O r =.] ^İ • \$ 9RS T 9. u] ^İ • 5 6. ' Š. 3ÜA " # [Session Server \] ^İ •](#) É Ê
M¢

Table 7. ECS%& Session Server 3 4 -] ^İ •

| < = e f | ' (| Sample g |
|--------------------|-----------------------|-------------|
| LENA_JVM_HEAP_SIZE | ¥ Heap Memory è % s O | 1024m (%!) |

| < = e f | ' (| Sample g |
|-------------------------------|--|-------------------------------|
| LENA_CONFIG_TEMPL ATE_ID | ¥ Service Cluster Ü : Revision No | SESSION-001 |
| LENA_MANAGER_ADD RESS | ¥ Service: Domain Î 1 : Uñ | lena- manager.local:7700 |
| JAVA_DOMAIN_CACHE _TTL | ¥ Domain Î 1 Cache ¾% (Ã) | 0 (%!) |
| LENA_SESSION_0_AD DRESS | ¥ Primary Session # O: # { Î 1, Service Ü > Service Discovery 3 O 5 x ë C² e Î 1 2 Ü 4 G f ô V | lena-session- 0.local:5180 |
| LENA_SESSION_1_AD DRESS | ¥ Secondary Session # O: # { Î 1, Service Ü > Service Discovery 3 O 5 x ë C² e Î 1 2 Ü 4 G f ô V | lena-session- 1.local:5180 |
| LENA_SESSION_EXPIR E_SEC | ¥ Session µ l ¾% (Ã) | 1800 (%!) |
| LENA_CONFIG_SHARE _SESSION | ¥ Application % Session F b – É | N |

5.4.3. Task ' 6

Task: † @, ±. —A Project: ~ P5 x y k j * %' ± N* ¼, 9 R 5 # \$ Container O: ù LENA
() L Ä 7 , - . É Ê µ 3 Ü. 9. * % 5 3 Ü G \$ 3 O: % P A 3 O a É Ê: 3 Ü L M ¢. 9.

Task 6U

Container Og' UV * \$ Task' O: . 9.

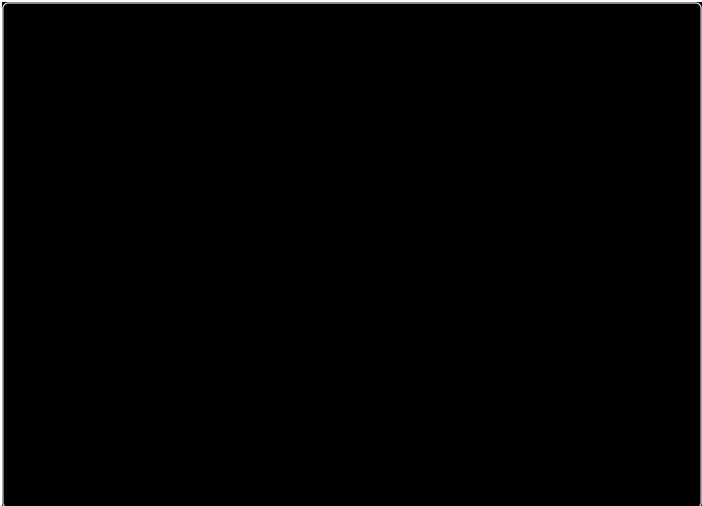


Figure 24. ECS%& Session Server 3 4 - Task O:

Container ' h

† @, † î s —: %! ContainerOg' k j . 9.



Figure 25. ECS%& Session Server 3 4 - Container -r

<=ef ' 6

Container 3O ù] ^İ • ' -r . 9.

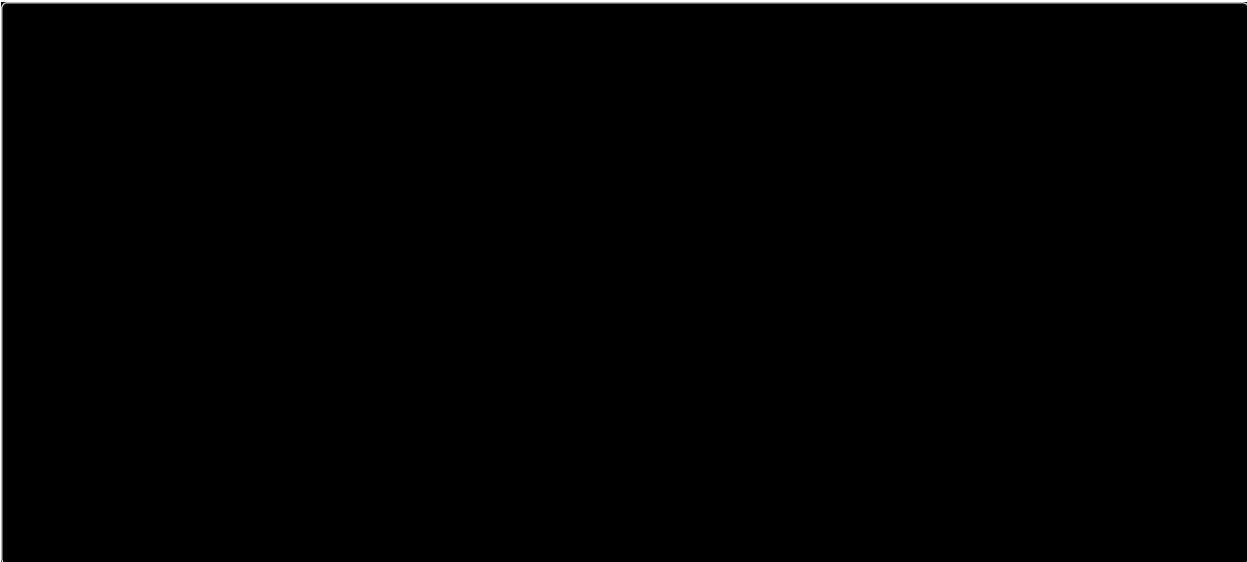


Figure 26. ECS%& Session Server 3 4 -] ^İ • 3O

■

LENA_SESSION_1_ADDRESS] ^İ • 5 &§¾ Secondary Session Service: Service
Discovery Î 1' k j . 9.

5.4.4. Service ' 6

|) 1 6U

+ # O: . Task' ^ E %» /() * % Ä. Service' O: . 9.



Figure 27. ECS%& Session Server 3 4 - Service O:

|) 1 • – (Service Discovery) ' 6

AWS\$ ECS% # { | û L " / * – ` @ ~ • t C Ž # { | L M (Service Discovery) %=L s^a . 9.
 Manager\$ Service Discovery %=L † @ * – N O o î 1' dg * – , P Server: 3 O h i > æ • Ž •
 %=L E F . 9.

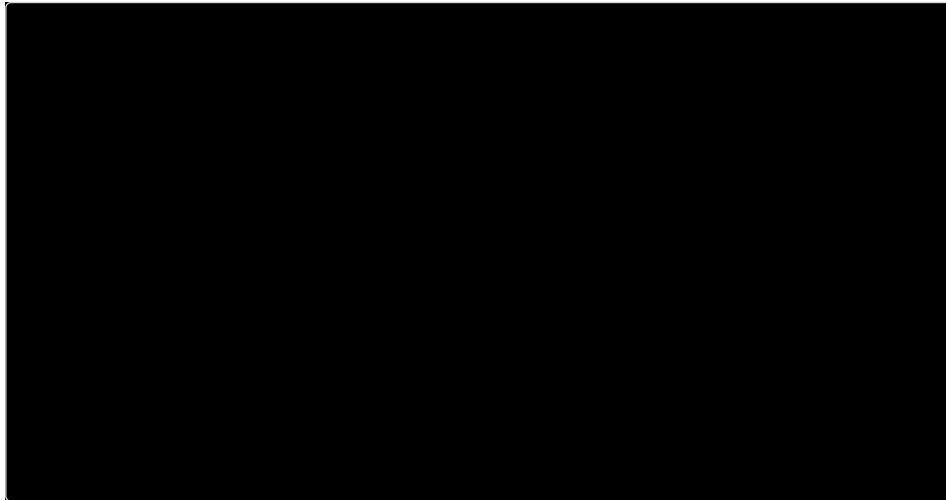


Figure 28. ECS%& Session Server 3 4 - Service Discovery 3 O

5.4.5. Service . — „ < æ

Service %» A Service O: ' Û Ç * • %» † ¾ ‡ o 9. ECS Cluster: Û • 5 # # { | > ‡ ´™ 5 #
 () ù e Service2 Task: ' " ' d e . 9

Service ~™ < æ



Figure 29. ECS%& Session Server 3 4 - Service ' " d e

Task ~ ™ < Œ

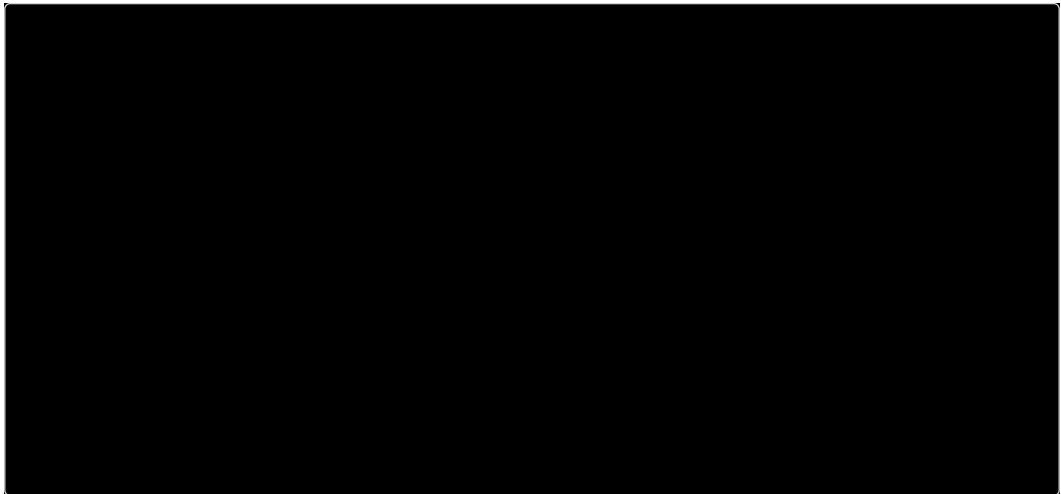


Figure 30. ECS%& Session Server 3 4 - Task ' " d e

5.5. WAS \]

5.5.1. \] x)

! " # WAS xUP{ ÉÊ 3ÜL M¢. 9.

5.5.2. ' 6 %&

. / ' 6 %&

WAS Container\$ 9RS TA ±N 3OL %PQD xU o9..

Table 8. ECS%& WAS 3 4 - xU %P

| ' 6 t • %& | ' 6 g / ' (|) * |
|----------------|-------------|-----|
| Service pq | Replica | |
| Container Port | TCP : 8180 | - |

ZXOu 76 %&

\@ ¾° Project] ^5 xy /O7# \@7ô ~ 3O - 12 †ö 3ÜG\$ Task > Service 3O5
\@o Sample çA 9RS T9.

Table 9. ECS%& WAS 3 4 - \ @¾° / O a

| ' 6 t • %& | ' (| Sample g |
|------------------------|---|-----------------------------------|
| Container Image | ProjectB Architecture / O5 x y , B o OS > JDK O; 5 R\$ LENA Manager Image | lenacloud/lena-cluster:{TAG_NAME} |
| Task > Service name | Task > Service : † @ | lena-was |
| label | Service2: `` / , L M5 ` @G\$ LabelID Key: Value– QD O: o 9. | type: lena-was |
| Service Namespace | ServiceL M¾ Domain Î 1 : SuffixD ` @ | local |
| Service Discovery Name | ServiceL M¾ Domain Î 1 : PrefixD ` @ | lena-was |
| Probe (Health Check) | < è Page, ¾± Delay ¾%, Î %\$ Application Pn5 RCž 3 O , - | 0/0 p 5 |

3 O r = .] ^İ • \$ 9RS T 9. u] ^İ • 5 6. ' Š. 3ÜA ! " # WAS] ^İ • ÉÊL MΦ. 9.

Table 10. ECS%& WAS 3 4 -] ^İ •

| < = e f | ' (| Sample g |
|-----------------------------------|--|--|
| LENA_CONFIG_TEMPL ATE_ID | ¥ Service Cluster Ü : Revision No | WAS-001:1 |
| LENA_MANAGER_ADD RESS | ¥ Service: Domain Î 1 : Uñ (+ # 3 4 o Manager: Service Î 1) | lena- manager.local:7700 |
| LENA_MANAGER_KEY | ¥ LENA_MANAGER_KEY : Open APID Manager Ö...¾ , - . e % c s ¥ Manager: Admin > Users ² J 5 # de r = | (À B Manager: Administration > IAM > Users ² J 5 # de > k j , -) |
| LENA_CONFIG_TEMPL ATE_DOWNLOAD | ¥ 3 O Og 9 (D\$ – É | Y |
| LENA_CONTRACT_CO DE | ¥ y † , 9 (D\$ ' Ä. © B H\$ ¥ y † , - - ¾ E F o H\$ ç | (À B H\$ de , -) |
| LENA_LICENSE_DOWN LOAD_URL | | manager |
| JAVA_DOMAIN_CACHE _TTL | ¥ Domain Î 1 Cache ¾% (Ä) | 3 (%!) |

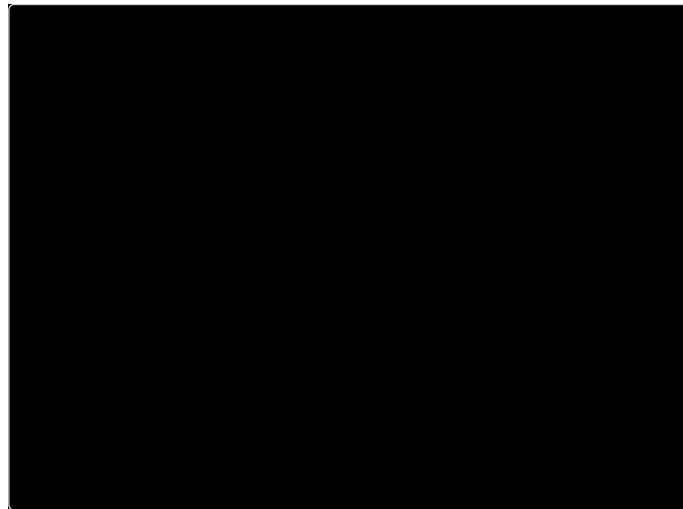
5.5.3. Task ' 6

Task: † @, ±. —A Project: ~ P5 x y k j * %' ± N * ¼, * % 5 # \$ Container O: ù LENA
() L Ä 7 , - . ÉÊµ 3Ü. 9. * % 5 3ÜG\$ 3O: %PA ! " # 3 O a ÉÊ: 3ÜL
MΦ. 9.

Task 6U

ECS%& WAS 3 4 - TaskO:

Container Og' UV* \$ Task' O: . 9.



Container ' h

†@,†î s—: %! ContainerOg' k j . 9.

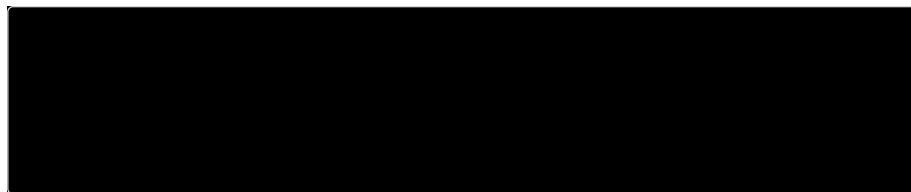


Figure 31. ECS%& WAS 3 4 - Container ¬r

<=ef ' 6

Container 3Où] ^ï • ' ¬r . 9.

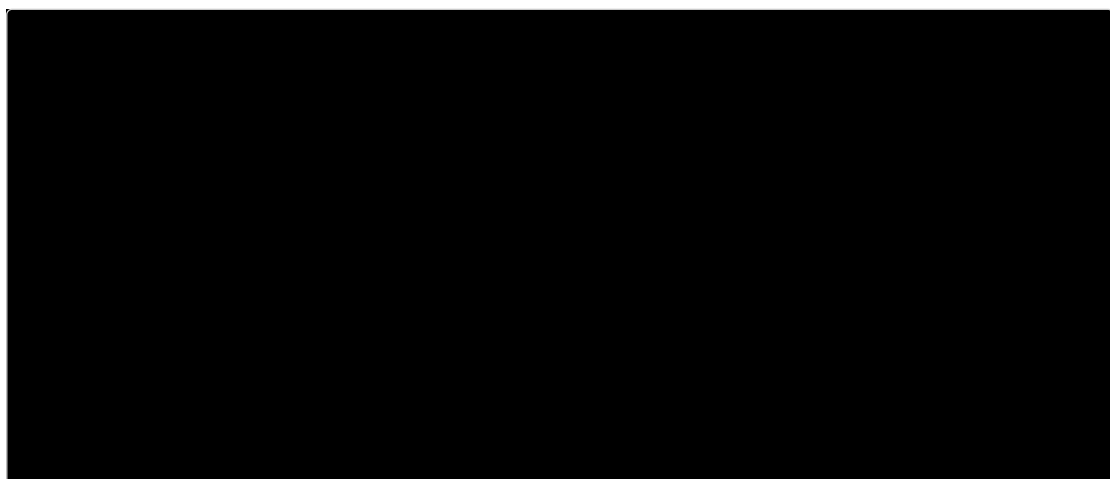


Figure 32. ECS%& WAS 3 4 -] ^ï • 3O

5.5.4. Service ' 6

|) 1 6U

+ # O: . Task' ^ E %» / () * % Ä. Service' O: . 9.

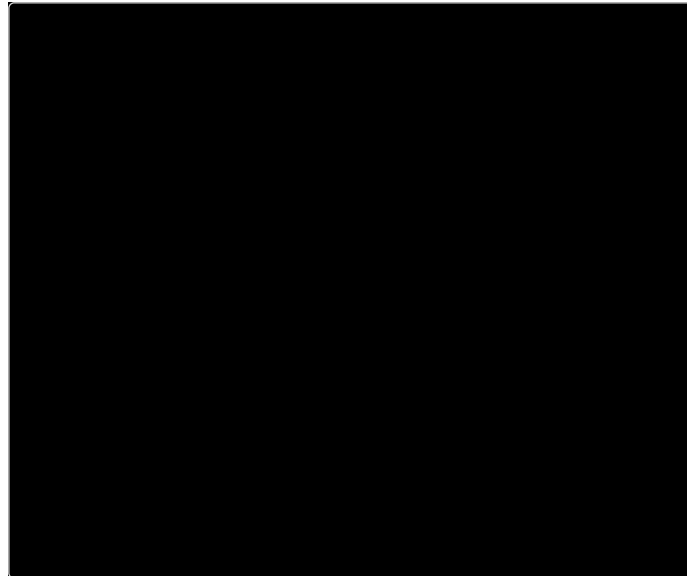


Figure 33. ECS%& WAS 34 - Service O:

|) 1 • – (Service Discovery) ' 6

AWS\$ ECS% # { | û L `` / * – ` @ ~ • t Cž # { | L M (Service Discovery) %=L s^a . 9.
Manager\$ Service Discovery %=L † @ * – NOo Î 1' dg* –, P Server: 3O hi > Æ • Ž •
%=L EF. 9.

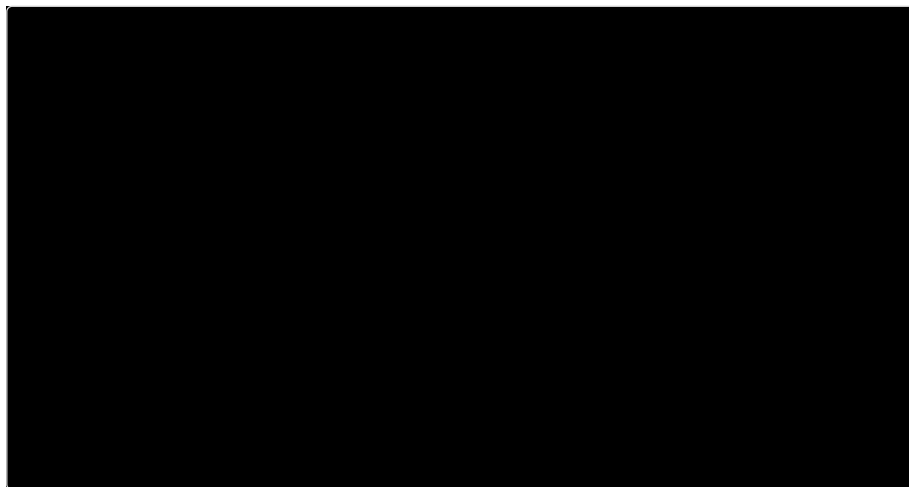


Figure 34. ECS%& WAS 34 - Service Discovery 3O

5.5.5. Service . — „ < Æ

Service %» A Service O: ' ÚÇ* • %» † ¾ ‡ o 9. ECS Cluster: Û • 5# # { | > ‡ ´™ 5#
() ù e Service2 Task: ' " ' de. 9.

Service ~™ < Æ



Figure 35. ECS%& WAS 3 4 - Service ' " d e

Task ~ ™ < Æ

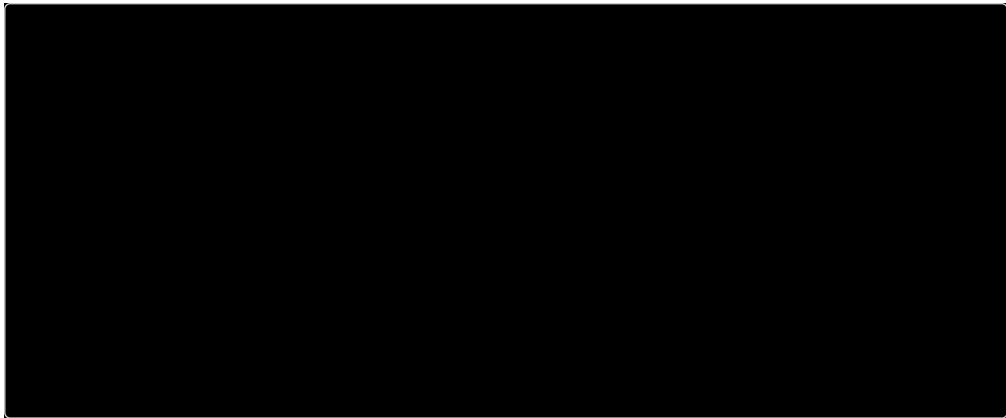


Figure 36. ECS%& WAS 3 4 - Task ' " d e

5.6. Embedded WAS \]

5.6.1. \] ×)

! " # [Embedded WAS ×UP{](#) É Ê 3ÜL M¢. 9.

5.6.2. ' 6 %&

. / ' 6 %&

Embedded WAS Container\$ 9RS TA ±N 3OL %PQD ×U o9..

Table 11. ECS%& Embedded WAS 3 4 - ×U %P

| ' 6 t • %& | ' 6 g / ' (|) * |
|----------------|-------------|-----|
| Service p q | Replica | |
| Container Port | TCP : 8180 | - |

ZXOu 76 %&

\@ ¾° Project] ^5 x y / O7# \@7ô ~ 3O - 12 † ö 3ÜG\$ Task > Service 3O5
\@o Sample çA 9RS T9.

Table 12. ECS%& WAS 3 4 - \@¾° / O a

| ' 6 t • %& | ' (| Sample g |
|------------------------|---|------------------------------------|
| Container Image | ProjectB Architecture / O5 x y , B o OS > JDK O; 5 R\$ LENA Manager Image | lenacloud/lena-embedded:{TAG_NAME} |
| Task > Service name | Task > Service : † @ | lena-was |
| label | Service2: `` / , L M5 ` @G\$ LabelD Key: Value– QD O: o 9. | type: lena-was |
| Service Namespace | ServiceL M¾ Domain Î 1 : SuffixD ` @ | local |
| Service Discovery Name | ServiceL M¾ Domain Î 1 : PrefixD ` @ | lena-was |
| Probe (Health Check) | < è Page, ¾± Delay ¾%, Î %\$ Application Pn5 RCž 3 O , - | 0/0 p 5 |

3 O r =.] ^İ • \$ 9RS T 9. u] ^İ • 5 6. ' Š. 3ÜA ! " # [Embedded WAS](#)] ^İ • ÉÊL MΦ. 9.

| < = e f | ' (| Sample g |
|------------------------------|--|---|
| LENA_CONFIG_TEMPLATE_ID | ¥ Service Cluster Ü : Revision No | WAS-001:1 |
| LENA_MANAGER_ADDRESS | ¥ Service: Domain Î 1 : Uñ (+ # 3 4 o Manager: Service Î 1) | lena-manager.default.svc.cluster.local:7700 |
| LENA_MANAGER_MONITORING_PORT | ¥ Manager Ė • Ž • Port Og | 16100 |
| LENA_APP_FILE | ¥ Application Jar ØÛ Ü | sample-app.jar |
| LENA_APP_DIR | ¥ Application Jar a b c i Ü | /usr/local/lena |

5.6.3. Task ' 6

Task: † @, ±. —A Project: ~ P5 x y k i * %' ±N* ¼, * %5# \$ Container O: ù LENA () L Ä7 , - . ÉÊµ 3Ü. 9. * %5 3ÜG\$ 3O: %PA ! " # [3O a](#) ÉÊ: 3ÜL MΦ. 9.

Task6U

ECS%& Embedded WAS 3 4 - TaskO:
Container Og' UV* \$ Task' O: . 9.



Container ' h

†@,†î s—: %! ContainerOg' k_j . 9.



Figure 37. ECS%& Embedded WAS 3 4 - Container ¬r

<=ef ' 6

Container 3Où] ^ï • ' ¬r . 9.

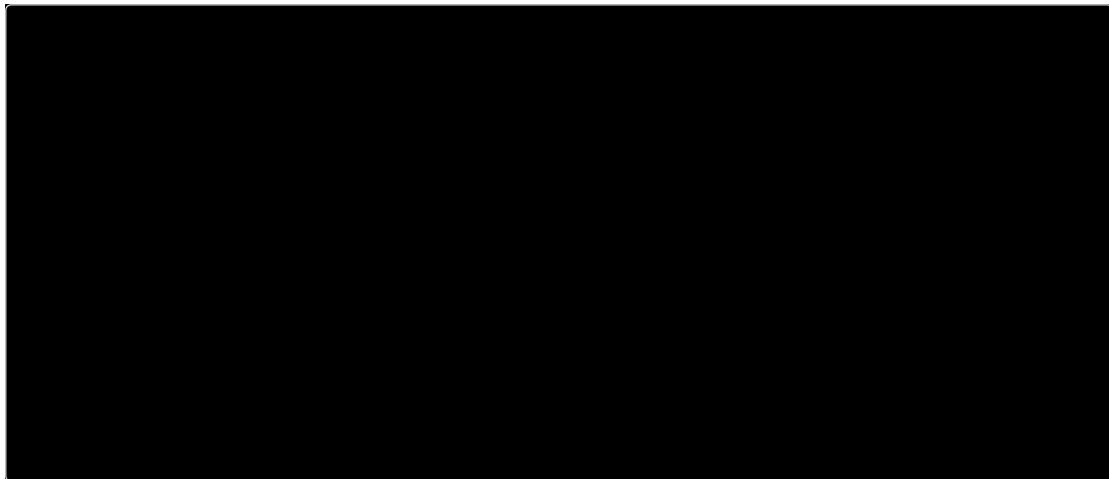


Figure 38. ECS%& Embedded WAS 3 4 -] ^ï • 3O

5.6.4. Service ' 6

|) 1 6U

+ # O: . Task' ^ E %» /() * % Ä. Service' O: . 9.

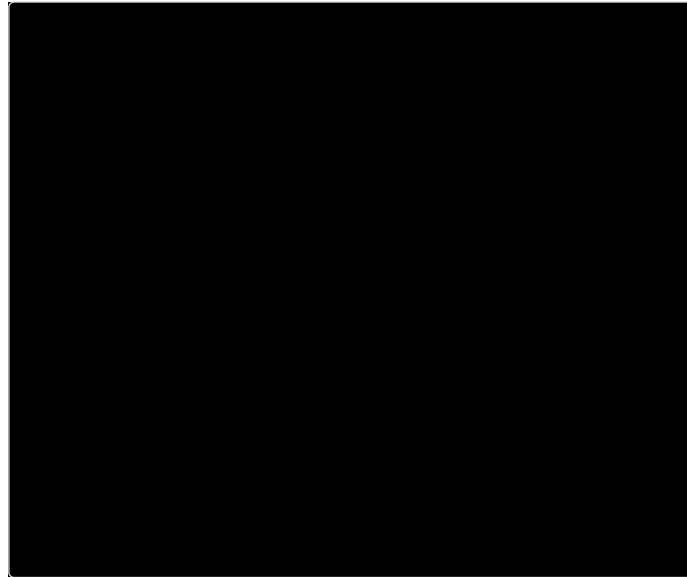


Figure 39. ECS%& WAS 3 4 - Service O:

|) 1 • – (Service Discovery) ' 6

AWS\$ ECS% # { | ũL ¨ / * – ` @ ~ • t Cž # { | LM (Service Discovery) %=L s^a . 9.
 Manager\$ Service Discovery %=L †@* – NOo Î 1' dg* –, P Server: 3O hi > Æ• Ž•
 %=L EF . 9.

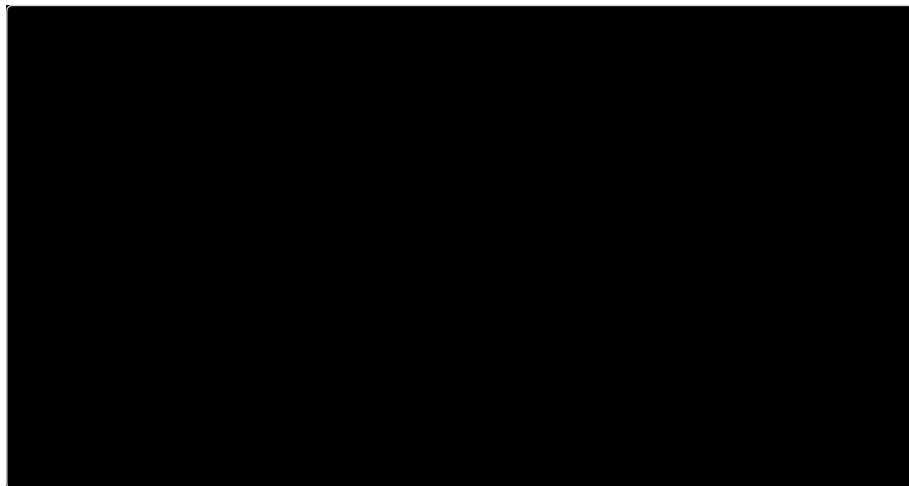


Figure 40. ECS%& Embedded WAS 3 4 - Service Discovery 3 O

5.6.5. Service . — „ < Æ

Service %» A Service O: ' ŪÇ* • %» † ¾‡o9. ECS Cluster: Ū• 5# # { | > ‡ ´TM5#
 () ùe Service2 Task: ' " ' de . 9.

Service ~TM < Æ



Figure 41. ECS%& WAS 3 4 - Service ' " de

Task ~™ < Æ



Figure 42. ECS%& WAS 3 4 - Task ' " de

5.7. Web Server \]

5.7.1. \] ×)

! " # [Web Server × U P{](#) ÉÊ: 3ÜL M¢. 9.

5.7.2. ' 6 %&

. / ' 6 %&

Web Server Container\$ 9RS TA ±N 3OL %PQD ×UGf ô . 9.

Table 13. ECS%& Web Server 3 4 - × U %P

| ' 6 t• %& | ' 6 g / ' (|) * |
|----------------|-------------|-----|
| Service pq | Replica | - |
| Container Port | TCP : 7180 | - |

ZXOu 76 %&

\@ ¾° Project] ^5 xy /O7# \@7ô ~ 3O - 12 †ö 3ÜG\$ ECS Task > Service 3O5 \@o Sample çA 9RS T9.

Table 14. ECS%& Web Server 3 4 - \@¾° /O a

| ' 6 t • %& | ' (| Sample g |
|------------------------|---|-------------------------------|
| Container Image | ProjectB Architecture / O5 x y , B o OS > JDK O; 5 R\$ LENA Manager Image | lenacloud/lena-web:{TAG_NAME} |
| Task > Service name | Task: † @, † ç A Task † @ / Hostname: PrefixD ` @ o 9. | lena-web |
| replica Å • | Container (Task) Å • | 2 |
| Probe (Health Check) | ¾ ‡ Delay¾%, Î %\$ Application Þ n 5 Rj 3 O , - | 0/0 p 5 |
| Service Namespace | ServiceL M¾ Domain Î 1 : SuffixD ` @ | local |
| Service Discovery Name | ServiceL M¾ Domain Î 1 : PrefixD ` @ | lena-web |

3 O r = .] ^İ • \$ 9RS T 9. u] ^İ • 5 6. ' Š. 3ÜA ! " # [Web Server](#)] ^İ • ÉÊL MΦ. 9.

Table 15. ECS%& Web Server 3 4 -] ^İ •

| < = e f | ' (| Sample g |
|-----------------------------------|--|-------------------------------|
| LENA_CONFIG_TEMPL ATE_ID | ¥ Service Cluster Ü : Revision No ¥ Revision Nor ž ç e ^ý Default RevisionL 9(D\$ ÅR | WEB-001:1 |
| LENA_MANAGER_ADD RESS | ¥ Service: Domain Î 1 : Uñ (+ # 3 4 o Manager: Service Î 1) | lena- manager.local:7700 |
| LENA_MANAGER_KEY | ¥ LENA_MANAGER_KEY : Open APID Manager Ö...¾, - . e %oCS ¥ Manager: Admin > Users ² J 5# der = | (ÅB Manager5# de k j , -) |
| LENA_CONFIG_TEMPL ATE_DOWNLOAD | ¥ 3 O Og 9(D\$ – É | Y |
| LENA_CONTRACT_CO DE | ¥ y † , 9(D\$' Ä. ©B H\$ ¥ y † , - - ¾ EF o H\$ ç | (ÅB H\$ de , -) |
| LENA_LICENSE_DOWN LOAD_URL | ¥ y † , 9(D\$ Ä4 | manager |
| LENA_RUN_AGENT | ¥ Agent ^ € – É | Y |

5.7.3. Task ' 6

Task: † @, ±. —A Project: ~ P5 x y k j * N, Container O: ù LENA () L Ä 7 , - . ÉÊµ
3Ü. 9. * %5 3ÜG\$ 3O: %PA ! " # [3Oa](#) ÉÊ: 3ÜL MΦ. 9.

Task 6U

Container Og' UV* \$ Task' O: . 9.

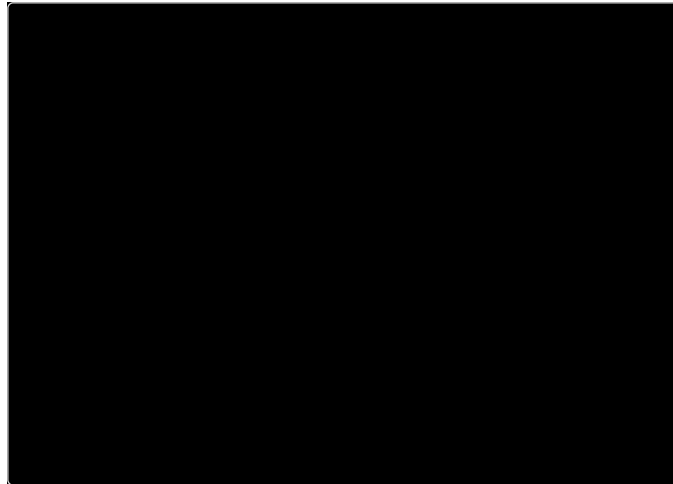


Figure 43. ECS%& Web Server 3 4 - Task O:

Container ' h

†@, †î s—: %! ContainerOg' k_i . 9

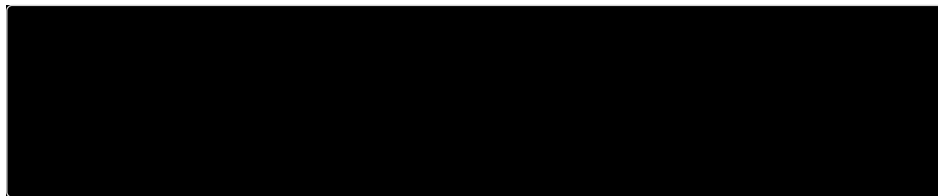


Figure 44. ECS%& Web Server 3 4 - Container ¬r

<=ef ' 6

Container 3Où] ^ï • ' ¬r . 9.

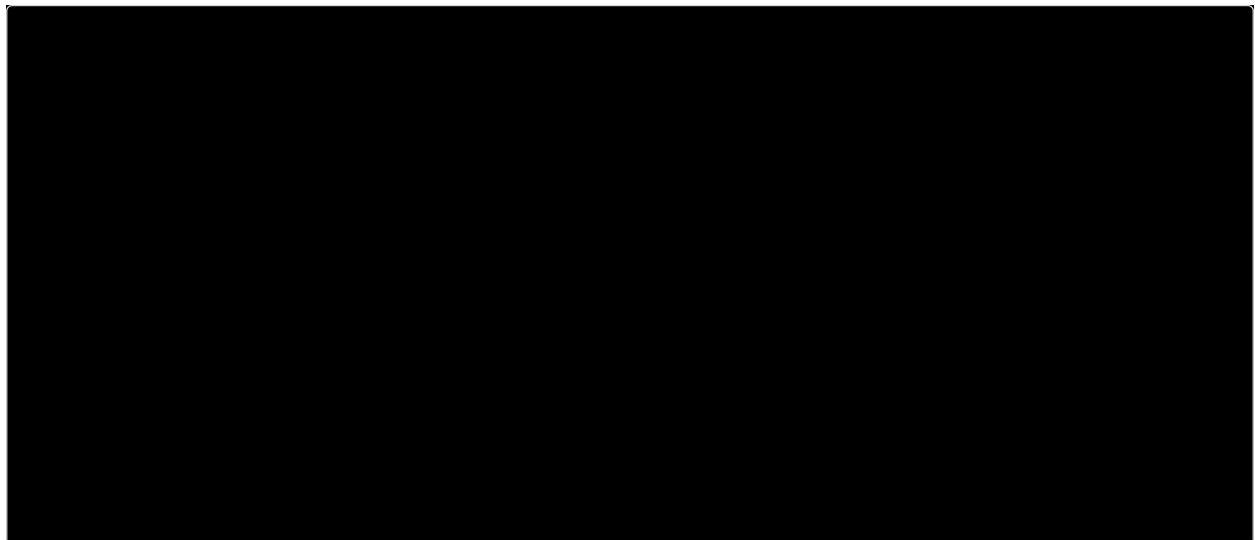


Figure 45. ECS%& Web Server 3 4 -] ^ï • 3O

5.7.4. Service ' 6

|) 1 6U

+ # O: . Task' ^ E %» /() * % Ä. Service' O: . 9.

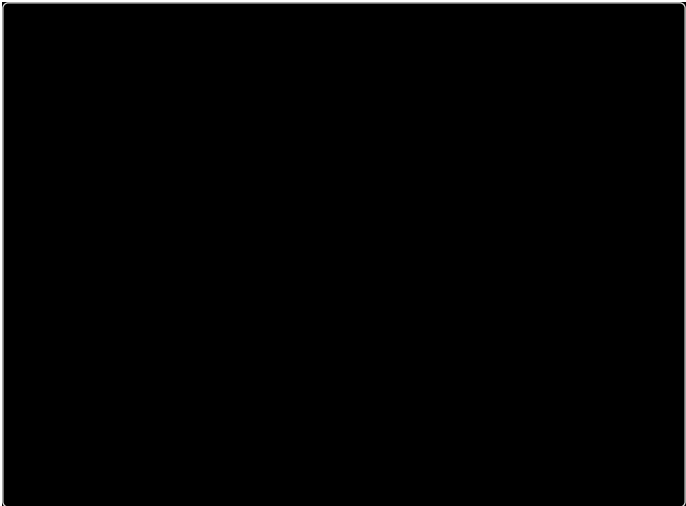


Figure 46. ECS%& Web Server 3 4 - Service O:

|) 1 • – (Service Discovery) ' 6

Web Server\$ ELB ' \$ Service Discovery %=L †@* – NOo Î 1' dg* – ÂÉ ' \$ P #{ | 5
Web #{ | ' EF ~ • t 9.



Figure 47. ECS%& Web Server 3 4 - Service Discovery 3 O

5.7.5. Service . — „ < Œ

Service %» A Service O: ' ÚÇ* • %» † ¾‡ o 9. ECS Cluster: Ũ• 5# #{ | > ‡ ´ ™5#
() ùe Service2 Task: ' " ' de. 9.

Service ~ ™ < Œ

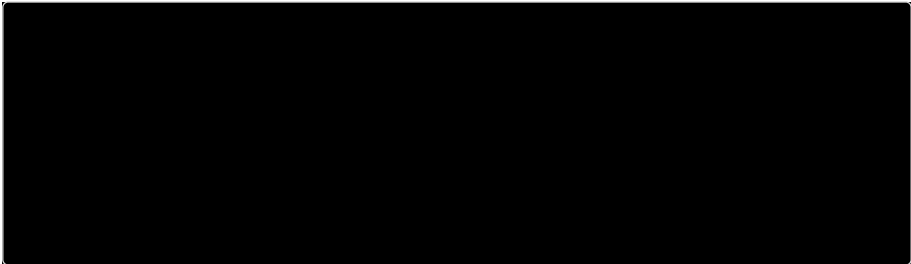


Figure 48. ECS%& Web Server 3 4 - Service ' " de

Task ~™ < Æ



Figure 49. ECS%& Web Server 3 4 - Task ' " de

Chapter 6. VM/Host . } ' s

6.1. ' s x)

34 P{ ‡ ´ QD 34ØÛL 6' #O5 ´ D\$ * N, Manager > Node Agent' 34 > ^ €. 9. † ö
34 ‡ ´ A Manager: Web UI' k 7# 34r r = * ¼, . ÛŸ —QD Web UI ÒÐ† Gs ½A ^ý,
ò \$ yeQDC » Û* j 34r r = * 9.

6.2. LENA ' s

LENA 34ØÛA gzipc , QD EFG¼, 34 6' #O5 ´ D\$ ö 34 j a b c i (\${LENA_HOME}
)5 UEL 7E. 9. %! 34 ^D\$ ð/engn001/lena/1.3/Ÿ ` @. 9.

LENA 34

```
[engn001]#  
[engn001]# tar -xzvf lena-1.3.x.tar.gz
```

34 ÆÈA @C5 xy 9RS T† EF† o9.

Table 16. LENA 34 ÆÈ ž

| Scripts | ' (|) * |
|------------------------------|---|------------------------------|
| lena-[O;].tar.gz | Web/Application k l 34 ØÛD Application Server, Web Server, Session Server 34 ÆÈ † ÆW UV | lena-1.3.x.tar.gz |
| lena-enterprise-[O;].tar.gz | Enterprise O; : WAS34 ÆÈ Enterprise O; 5\$ Session Serverr UV | lena-enterprise-1.3.x.tar.gz |
| lena-standard-[O;].tar.gz | Standard O; : WAS34 ÆÈ | lena-standard-1.3.x.tar.gz |

6.3. k l mn ! "

LENA 34' Ä. ØÛL P{ . 9. LENA 34 ØÛA BCD EF o9.

\${LENA_HOME}: a b c i m¢\$ Wv2 T9.

Table 17. a b c i m¢

| k l mn | ' (|) * |
|--------|--|-----|
| bin | Node Agent2 Manager: Start/Stop scripts, install scripts E F | |
| conf | Node Agent, Manager —: 3OØÛ | |

| k l mn | ' (|) * |
|----------|---|-----|
| database | Œ•Ž•5# ° n. ÛB " †Ž' ÚÇ* \$ a b c i | |
| depot | 3 4' Ä. Local Repository | |
| etc | %P ² P Og > 3 O ØÛ | |
| license | License Og' hi * \$ a b c i | |
| logs | Node Agent / Manager DœØÛ | |
| modules | ^ €5 , - . ŒË† Ä4* \$ ^ D (lena-node-agent, lena-installer, lena-manager —) | |
| servers | Serverr 3 4â %! ^ D | |
| tmp | l ¾a b c i | |

E F * \$ ^ € Scripts \$ Wv 2 T 9. (\${LENA_HOME}/bin 5 Ä 4)

Table 18. E F Script ž

| Scripts | ' (|) * |
|------------------------|---|----------------------------|
| install.sh | #O' 3 4* % Ä. %! script | |
| web-compile.sh | Web Server' y ØÛ* % Ä. script | |
| web-package-install.sh | Web Server y ØÛ > m» 5 , - . QXs 3 4 script | Linux only, root ±. , - |
| crypt.sh | Datasource5 ` @* \$ Password • » v p Û ^ € (k i . " H¢L v p Û " H¢D Ĩ]) | |
| env-manager.sh | Manager^ €L Ä.] ^ Ĩ • | Manager 3 4¾ |
| start-manager.sh | Manager: ^ € | Manager 3 4¾ |
| stop-manager.sh | Manager: p l | Manager 3 4¾ |
| ps-manager.sh | Manager: ĩ DŠ de | Manager 3 4¾ |
| start-agent.sh | Node Agent: ^ € | |
| stop-agent.sh | Node Agent: p l | |
| ps-agent.sh | Node Agent: ĩ DŠ de | |

] ^ 3 O ØÛA Wv 2 T 9. (\${LENA_HOME}/conf 5 Ä 4)

Table 19.] ^ 3 O ØÛ ž

| Config File | ' (|) * |
|--------------|-------------------|-----|
| manager.conf | Manager h (30 | |
| agent.conf | Node Agent h (30 | |

6.4. Manager ' s

EFG\$ LENA\$ Web Server, WAS2 Session Server, Node/Server5 34Gf Ef > Status'
de* \$ Agent2 hi H5j EFG\$ ManagerD mno9.

6.4.1. Manager ' s

Manager\$ install.shL †@* – Wv2 TA • #D 34. 9.

1. \${LENA_HOME}/bin/install.sh create lena-manager
2. Service Port Og' k j . 9. (default: 7700)
3. #O ' " Og' • ¿ ÁL port Og' k j . 9. %! 3OL ` @* ¼, Manager' -r D 34* \$
^ý5\$ port' İ ^ . 9. (default: 16100)
4. Manager' ^ € ~ OS©OL k j . 9. (default: | èpñ ^ € bÚ)

LENA Manager 34

```
[bin]$ ./install.sh create lena-manager
*****
* LENA Server Install !          *
*****

+-----+
+-----+
| 1. SERVICE_PORT is the port number used by Manager.
|    ex : 7700
| 2. MONITORING_PORT is the port number used by Manager for monitoring.
|    ex : 16100
| 3. RUN_USER is user running Argo Manager.
|    ex : tomat
+-----+
+-----+

Input SERVICE_PORT for installation. (q: quit)
Default value is '7700'

Input MONITORING_PORT for installation. (q: quit)
Default value is '16100'

Input RUN_USER for installation. (q: quit)
Default value is 'lena'

===== Execution Result =====
LENA_HOME : /engn001/lena/1.3
JAVA_HOME : /engn001/java/jdk1.8.0_191
SERVER_ID : lena-manager
SERVICE_PORT : 7700
MONITORING_PORT : 16100
INSTALL_PATH : /engn001/lena/1.3/modules/lena-manager
RESULT : Success
MESSAGE : create succeeded
=====

Execution is completed.!!
[bin]$
```

!

- • 6: Ç{ D #{ | ' * \$ ^ý, Manager\$. 6: Ç{ 5µ 34. 9.

6.4.2. Manager VW

Manager' %» * – O' \QD 34G` \$s de. 9.

1. start-manager.sh ØÛL ^ €. 9.

LENA Manager ^ €

```
[bi n]$ ./start-manager.sh
-----
Ê          LENA Manager
-----
Usi ng LENA_HOME :    /engn001/I ena/1.3
Usi ng JRE_HOME :    /engn001/j ava/j dk1.8.0_191
Usi ng SERVER_PID :   /engn001/I ena/1.3/modul es/I ena-manager/I ena-
manager_sol manager. pi d
Usi ng SERVER_HOME :  /engn001/I ena/1.3/modul es/I ena-manager
Usi ng SERVER_ID :    I ena-manager
Usi ng INSTANCE_NAME : I ena-manager_sol manager
LENA started.
[bi n]$
```

2. [http://\[Manager IP\]:7700](http://[Manager IP]:7700) 5 ÒÐ* – Wv J † s' de. 9.(Ã%Ç: admin/!admin1234)

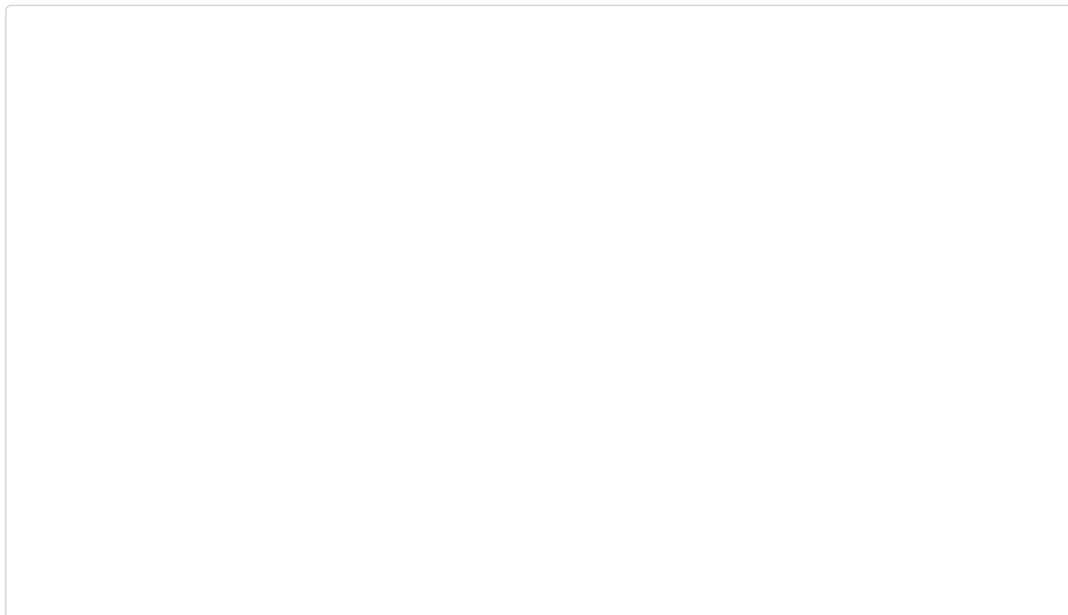


Figure 50. LENA Manager Døe

3. stop-manager.sh ØÛL ^ €* – pl ~ • t 9.

LENA Manager p1

```
[bi n]$ ./stop-manager.sh
-----
Ê          LENA Manager
-----
Usi ng LENA_HOME : /engn001/l ena/1.3
Usi ng JRE_HOME: /engn001/j ava/j dk1.8.0_191
Usi ng SERVER_PID: /engn001/l ena/1.3/modul es/l ena-manager/l ena-
manager_sol manager. pi d
Usi ng SERVER_HOME : /engn001/l ena/1.3/modul es/l ena-manager
Usi ng SERVER_ID : l ena-manager
Usi ng INSTANCE_NAME : l ena-manager_sol manager
LENA stopped.
##### l ena-manager_sol manager successfully shut down #####
[bi n]$
```

6.5. Node Agent VW

Node Agent\$ Node, Server: Ef > Œ• Ž• %=L • ' * \$ Agent † 9. Node Agent\$ LENA34¾
 %! \QD 34r G¼, Node5 6. Og' r, <% Ä. Agent' ^ €* – Ö . 9. Node Agent\$
 Web/Application/Session Server: ' " ŒŒ > ¾† S pl ' • € ~ • t 9.

6.5.1. Node Agent VW

\${LENA_HOME}/bin/start-agent.sh ØÛL ^ €. 9. JAVA_HOME† sOGs ½A ^ý, terminal5#
 JAVA_HOMEL k; * y\$ ²¾sr Í <j o9.†), JAVA_HOME: ^D' k; * • agentr ^ €o9.

```
[bin]# ./start-agent.sh
Input JAVA_HOME path for LENA. ( q: quit )
JAVA_HOME PATH :
/engn001/j ava/j dk1.8.0_191
Input Agent port for LENA Agent. ( q: quit )
Agent port (Default : 16800):
Input Agent user for LENA Agent. ( q: quit )
Agent user (Default : root):
root

-----
Ê          LENA Agent
-----

Using LENA_HOME : /engn001/l ena/1.3
Using JAVA_HOME : /engn001/j ava/j dk1.8.0_191/j re
Using CONF_FILE : /engn001/l ena/1.3/conf/agent.conf
Using LOG_HOME : /engn001/l ena/1.3/l ogs/l ena-agent
Using RUN_USER : root
Using PORT : 16800
Using UUID : d03ddd60-de12-35df-9ea1-a409a3085eeb
LENA Agent is started.
[bin]#
```

6.5.2. Node Agent —z [l < Œ

`${LENA_HOME}/bin/ps-agent.sh ØÛL ^ €* – Wv 2 T † Process: ' " ' de. 9.`

```
[bin]$ ./ps-agent.sh
l ena      24208      1 62 14:00 ?      00:00:03
/engn001/j ava/j dk1.8.0_191/bi n/j ava -Xms64m -Xmx256m
-Dl ena.home=/engn001/l ena/1.3 -Dl og.home=/engn001/l ena/1.3/l ogs/l ena-agent
-Dpatch.l og.home=/engn001/l ena/1.3/l ogs/l ena-patcher
-Dj ava.l i brary.path=: /engn001/l ena/1.3/modul es/l ena-agent/l i b/si gar
-Dj ava.net.preferIPv4Stack=true -cp .: /engn001/l ena/1.3/modul es/l ena-
agent/l i b/bcprov-j dk15on-1.55.j ar: /engn001/l ena/1.3/modul es/l ena-
agent/l i b/l ena-agent-1.3.0.j ar: /engn001/l ena/1.3/modul es/l ena-
agent/l i b: /engn001/j ava/j dk1.8.0_191/l i b/tool s.j ar
argo.node.agent.server.NodeAgentServer -start
[bin]$
```

6.5.3. Node Agent GŠ

`stop-agent.sh' ^ €* – pl ~ • t 9.`

```
[bi n]$ ./stop-agent.sh
-----
Ê          LENA Agent
-----
Usi ng LENA_HOME : /engn001/l ena/1.3
Usi ng JAVA_HOME : /engn001/j ava/j dk1.8.0_191/j re
Usi ng CONF_FILE : /engn001/l ena/1.3/conf/agent.conf
Usi ng LOG_HOME : /engn001/l ena/1.3/l ogs/l ena-agent
Usi ng RUN_USER : l ena
Usi ng PORT : 16800
Usi ng UUID : 0d5f6a4a-1084-4bac-ad8c-70b67bf3e495
LENA Agent is stopped normally.
[bi n]$
```

6.6. Session Server ' s (WEB UI . })

Session Server' hi * % Ä. Û• L EF. 9. Node5 34. Session Server: —ž , • O, ŸEr
r = * ¼, ¾± S pl ShellL ^ € ~ • t 9.

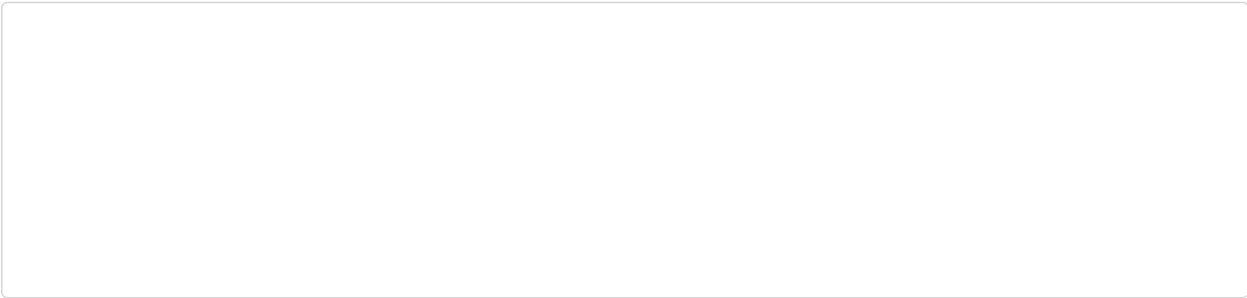


Figure 51. Session Server ž

Session Server: ÐnA Wv 2 T 9. (*) \$, • Ğ

Table 20. Session Server: Ðn

| %& | ' (|) * |
|-------------|--|-----|
| Status | Session Server: ' " | |
| Name(*) | Session Server: † @ | |
| IP(*) | Session Server: IPÎ 1 | |
| Server ID | Session Server: Identifier | |
| Port | Service Uñ t p | |
| Server Type | Session Server: bc | |
| Start/Stop | Server: ¾± > p l | |
| + > † œ | Register • • ' \$ f 6(P_) • • L — ~ * — , fo Server Ogr İ ^ ù l L ~ ¾ | |
| - > † œ | ‡ + (• LJ) • • L — ~ * — , fo ServerOgr ŸE „ L ~ ¾ | |

6.6.1. Session Server ' s

1. **Install** • • L —~ . 9.

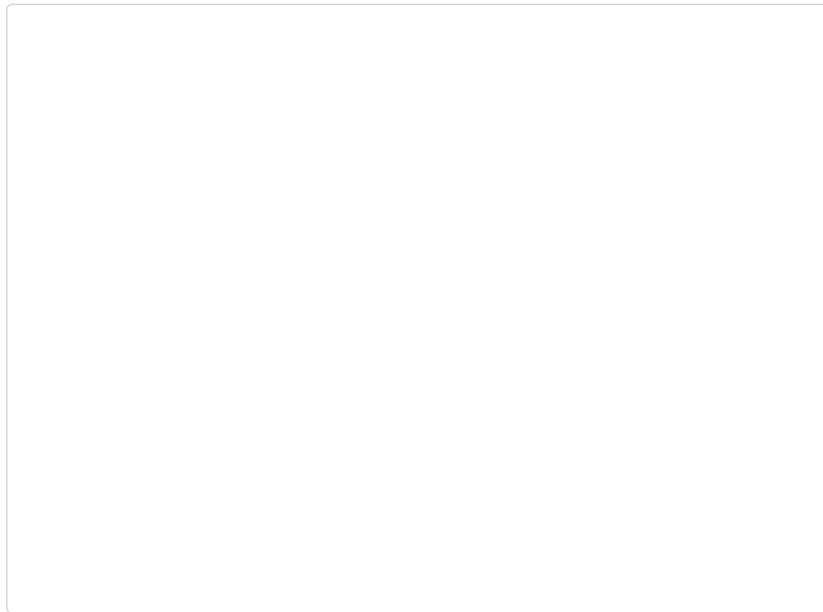


Figure 52. Session Server 34¾ k j Ũ•

2. Server ID2 Service Port, Secondary Server IP/Port' k j . 9.
3. 'Save' O£L —~ * — ÚÇ. 9.

!

```
¥ Node5 ^ E 34Gf t$ #O2 Manager5# hi *$ #O: Og5$ ' t r
t L • t 9. (console%& 34 ¾)

¥ #OIDù + <qr ~ ° * $ ^ ý, Register%=L t @* — 34o #OOg'
de. 9.

¥ Manager IP$ Node: host IPD H» k j o 9. " ñ#è mn5 xy H» k j o
IPr ^ E " ñ#è IP2 9è ^ ýr ~ ° ~ • t 9. t) $ Manager IP' • O* —
k j 7ô . 9.
```

6.6.2. Server VW

1. **Stop** • • L —~ * — Server' p l . 9.
2. **Start** • • L —~ * — Server' ¾±. 9.

!

¾± r =. ' " Û ^ ý 5µ ¾± O£ t Ån Û o 9.

6.6.3. Server ‡ +

1. **‡ + (• L J)** • • L —~ * — ServerOg' ÝE r =. ' " D Ĩ ^ . 9.
2. **Save** • • L —~ . 9.
3. OKO£L ÷ ì • Manager: DB" † Ž2 Oi \ #O' Š; ! ÝE * N, CancelO£L —~ * •
Manager: DB " † Žµ ÝE. 9.

6.6.4. Server • ,

Console %&QD 34. #O' Manager' k 7# hi * _• , Server Og —ž † , - * 9.

1. **+Register** • • L —~ . 9.
2. —ž ~ #O' —~ . 9.

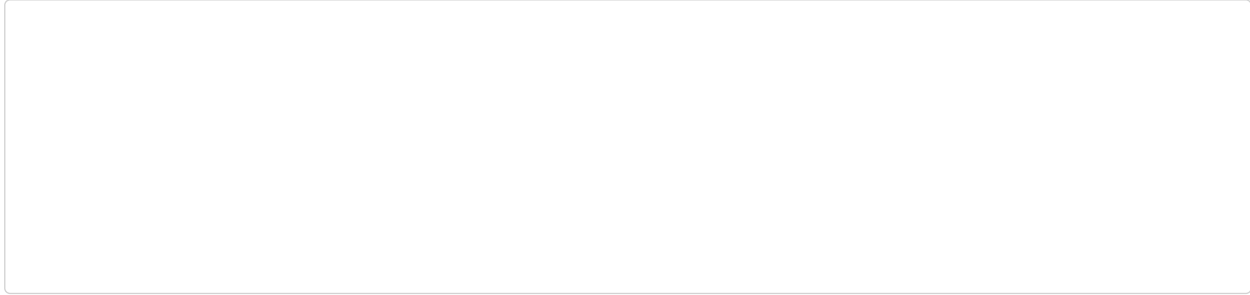


Figure 53. Session Server —ž ¾ Server , f Ů•

3. **Save** • • L —~ * — ÚÇ. 9..

6.7. Session Server ' s (CLI . })

6.7.1. Session Server ' s

Session Server\$ Embedded2 StandaloneO; QD mÊo9. Embedded O; : ^ý Application #O
 ?5 UVGf tf BC 34r , - * Q¼, Standalone O; 34 ¾ install.shL †@* — Wv2 TA
 • #D 34. 9.

1. \${LENA_HOME}/bin/install.sh create lena-session

Session Server 3 4

```
[bin]$ ./install.sh create lena-session
*****
* LENA Server Install !                *
*****
+-----+
+-----+
| 1. SERVER_ID means business code of system and its number of letter is
| from 3 to 5.
|   ex : tom1, tc01, svr01
| 2. SERVICE_PORT is the port number used by Session Server.
|   ex : 8080
| 3. SECONDARY_SERVER_IP is the ip number communicate with Secondary Session
| Server
|   ex : 127.0.0.1
| 4. SECONDARY_SERVICE_PORT is the port number used by Secondary Session
| Server.
|   ex : 8080
| 5. RUN_USER is user running Session Server
|   ex : tomat, apache
| 6. INSTALL_ROOT_PATH is is server root directory in filesystem.
|   ex : /ssw, /sw/server, /ssw/was
+-----+
+-----+
```

2. k j a

[a BD defaultç† ~ ¼G¼,İ ^† , - . ^ý ` @Hr ÑÒ k j * – İ ^ ~ • t 9.

Session Server 3 4¼ k j a K¼

```
Input SERVER_ID for installation. (q: quit)
tm-session1
Input SERVICE_PORT for installation. (q: quit)
Default value is '5000'
5005
Input SECONDARY_SERVER_IP for installation. (q: quit)
127.0.0.1
Input SECONDARY_SERVICE_PORT for installation. (q: quit)
Default value is '5001'
5006
Input RUN_USER for installation. (q: quit)
Default value is 'lena'

Input INSTALL_ROOT_PATH for installation. (q: quit)
Default value is '/engn001/l ena-1.3.0/tmservers'

Ê===== Execution Result =====
LENA_HOME : /engn001/l ena/1.3
JAVA_HOME : /engn001/j ava/j dk1.8.0_191/j re
SERVER_ID : tm-session1
SERVICE_PORT : 5005
SECONDARY_SERVER_IP : 127.0.01
SECONDARY_SERVICE_PORT : 5006
RUN_USER : l ena
INSTALL_PATH : /engn001/l ena/1.2/servers/session1
RESULT : Success
MESSAGE : create succeeded
Ê=====

create is completed.!!
[bi n]$
```

Table 21. Session Server 3 4¼ k j a

| %& | ' (|) * |
|------------------------|---------------------------|-----------------|
| SERVER_ID | Session Server: ID | |
| SERVICE_PORT | Session Server: #{ Uñ | Default: 050000 |
| SECONDARY_SERVER_IP | Secondary Server: IPÎ 1 | |
| SECONDARY_SERVICE_PORT | Secondary Server: #{ Uñ | Default: 050010 |

| %& | ' (|) * |
|-------------------|---------------------------------|---|
| RUN_USER | Session Server' ^ €* \$ ^ € ©OÜ | Default: ò èpñ ^ € ©OÜ |
| INSTALL_ROOT_PATH | Session Server' 34~ ' Ä a b c i | Default: 0\${LENA_HOME}/tmse rvers0 |

¥ \$INSTALL_ROOT_PATH/tmservers/0SERVICE_ID0 Directory° nL de. 9.

!

install.sh • € ¾ * Í : Session Serverr 34G¼, N Ä: #O 34 ¾ install.shL
NE • €7ô . 9.

6.7.2. Session Server VW

Session Server' %» * – O' \QD 34G` \$s de. 9.

1. Session Server 34 Ä45# start.sh ØÜL ^ €. 9.

Session Server %»

```
[tm-session1]$ ./start.sh
-----
Start Session Server
-----
Using LENA_HOME : /engn001/lena/1.3
Using SERVER_HOME : /engn001/lena/1.3/servers/tm-session1
Using SERVER_ID : tm-session1
Using JAVA_HOME : /engn001/java/jdk1.8.0_191

Session Server Started..
[tm-session1]$
```

2. ps.sh ØÜL ^ €* – ï DŠ | : ' " ' de. 9.

Session Server ï DŠ | ' " de

```
[tm-session1]$ ./ps.sh

lena 16232      1      1 09:56 pts/7 00:00:00
/engn001/java/jdk1.8.0_191/bin/java -Xmx1024m -Dzodiac.name=session_5105
-Dzodiac.logdir=/engn001/lena/1.3/logs/session-server -cp
.::/engn001/lena/leesyong/1.2/servers/tm-session1/lib/lena-session-common-
1.2.0.jar:/engn001/lena/leesyong/1.2/servers/tm-session1/lib/lena-session-
server-1.2.0.jar -Dzodiac.config=session.conf zodiac.server.Main

[tm-session1]$
```

Session Server pl

3. stop.sh ØÙL ^ €* – pl ~ • t 9.

```
[tm-sessi on1]$ ./stop.sh
-----
Stop Sessi on Server
-----
Usi ng LENA_HOME : /engn001/l ena/1.3
Usi ng SERVER_HOME : /engn001/l ena/1.3/servers/tm-sessi on1
Usi ng SERVER_ID : tm-sessi on1
Usi ng JAVA_HOME : /engn001/j ava/j dk1.8.0_191

Sessi on Server Stoped..
[tm-sessi on1]$
```

6.7.3. Session Server ‡ +

% 34o #O\$ | èpñ' †@* – Uninstall~ • t 9.

LENA5#\$ 34o #O: Og' BC: xmlØÙ5 ÚÇ* N t 9. xy#, directory' ÑÒ ÿE * s
½N, install.sh | èpñ' †@* – Uninstall 7Ô . 9.

1. install.sh | èpñ ^ €

- [Session Server : \${LENA_HOME}/bin/install.sh delete lena-session
- [Manager : \${LENA_HOME}/bin/install.sh delete lena-manager

Session Server ŸE

```
[Iena@RNDTOMCAT1V bin]$ ./install.sh delete tm-session
*****
* LENA Server Install ! *
*****

+-----+
+-----+
| 1. SERVER_ID : Server'id to delete
+-----+
+-----+

Input SERVER_ID for installation. (q:quit)
tm-session
===== Execution Result =====
LENA_HOME : /engn001/Iena/1.3
JAVA_HOME : /engn001/java/jdk1.8.0_191/jre
SERVER_ID : Ienawas2
DELETE_PATH : /engn001/Iena/1.3/servers/tm-session
RESULT : Success
MESSAGE : delete succeeded
=====

delete is completed.!!

[bin]$
```

2. k j a

Table 22. Session Server ŸE¾† j a

| %& | ' (|) * |
|-----------|-----------------------|--|
| SERVER_ID | Uninstall~ Server: ID | Manager: ^ý idr lena-managerD H» k j G¼, BCD Server ID' k j Ás ½\$9. |



LENA5#\$ 34o #O: Og' BC: xmlØÙ5 ÚÇ* N t 9. xy#, directory'
ÑÒ ŸE * s ½N, install.sh | èpñ' †@* – Uninstall7ô . 9.

Chapter 7. E ž

7.1. LENA L M SpecE • Ÿ

Table 23. LENA s^a Spec

| Specification | Version |) * |
|--|---------|-----|
| Java Development Kit (JDK) | 1.8~ | |
| Java Servlet | 3.1 | |
| Java Server Pages (JSP) | 2.3 | |
| Expression Language (EL) | 2.2 | |
| JavaServer Pages Standard Tag Library (JSTL) | 1.2 | |
| Enterprise JavaBeans (EJB) | 3.2 | |
| Java Message Service (JMS) | 1.1 | |
| Java Transaction API (JTA) | 1.2 | |
| Java API for RESTful Services (JAX-RS) | 2.0 | |
| Java API for XML Web Services (JAX-WS) | 2.2 | |

7.2. Manager DBv w %f

Manager: ?É" †Ž hi ' Ä. HSQL DB: ØÙA Î %\QD(1Ù) e´ØÙL ° n*N t9.
° nÄ4\$ \${LENA_HOME}/repository/backup/database †9.

%! \QD 30Ù †; e´Og\$ ŸE*Cž Gf t\$" gh%%L Ĩ ^*N ¨A ^ý,
\${LENA_HOME}/conf ¥F * Ä5 manager.conf ØÙL ¢N, dbbackup.size=gh%%L kĭ ö
Manager' È %» * • gh%%L Ĩ ^~ • t9.

7.3. Manager U I †r ‡ +

Managerr ?É\QD ĩ %\$ †ĭ A Î %\QD ŸE*Cž ĩ 3\$•† Gf t9. ŸE*\$ Og\$
Action Trace †ĭ S Server History †ĭ †9.

%! \QD Action Trace†ĭ A 30Ù] sµ gh*N, Server History †ĭ A 90Ù] s gh*N t9. †
gh%%L Ĩ ^*N ¨A ^ý \${LENA_HOME}/repository/conf ¥F * Ä5 manager.conf ØÙL ¢N,
actiontrace.size=gh%%, serverhistory.size=gh%%L kĭ ö Manager' È %» * • gh%%L
Ĩ ^~ • t9.

7.4. Manager U admin ĭ 1 c€ ¢. £

Manager: admin` @H Q| #Š' Ê^*-Í { ``tp <q©•r ÄS*†L ^ý5\$ console'
k*-Q| #Š' Ä%Û7ö . 9.

1. Managerr 34o Ç{ 5 console(telnet or ssh)D ØÐ. 9.
2. \$LENA_HOME/bin/reset_manager_pw.sh ØÙL ^€ . 9.

3. Q | # S' Å%Û ~ user adminL k j . 9.
4. Å%Û~ Q | # S' k j . 9. &, Q | # S \$ 8Hi t' , a Ø« /YH/P• " H: ¢I QD k j . 9.
Q | # S \$ g , L Ä7 console5 ~ ¾Gs ½\$9.

Manager: admin Q | # S Å%Û

```
[bin]$ ./reset-manager-pw.sh
*****
* LENA Server Install ! *
```

```
*****
+-----+
--
| 1. USER_ID is the user id to reset
| ex : admin
| 2. NEW_PASSWORD is the password to change
| - password rule #1 : more than 8 length
| - password rule #2 : inclusion of one or more alphabet characters
| - password rule #3 : inclusion of one or more numerical digits
| - password rule #4 : inclusion of one or more special characters
+-----+
--
Input USER_ID for installation. (q:quit)
administrator
Input NEW_PASSWORD for installation. (q:quit)

The password has been changed successfully.
Execution is completed.!!s
```

7.5. LENA ' s æ ¥ OSv | S '' (CentOS. x)

LENA 34 ¾ OSØyî Ž \$ max user processes ¢L 8192†' QD 3O* \$ êL ±Ç. 9.

Table 24. ±Ç OSØyî Ž (CentOS %P)

| parameter | æ ¥g | . /g |
|--------------------|------|------|
| max user processes | 8192 | 1024 |
| open files | 8192 | 1024 |

CentOS%PQD max user processes 3OA 9RS T† ůlimit ÐaŮ ÜÝf' ^ €* – deL ~ • t 9.

OS Øyî Ž max user processes de (CentOS %P)

```
$ ulimit -a +
core file size (blocks, -c) 0 +
data seg size (kbytes, -d) unlimited +
scheduling priority (nice, -e) 0 +
file size (blocks, -f) 8192 +
pending signals (-i) 14891 +
max locked memory (kbytes, -l) 64 +
max memory size (kbytes, -m) unlimited +
open files (-n) 1024 +
pipe size (512 bytes, -p) 8 +
POSIX message queues (bytes, -q) 819200 +
real-time priority (-r) 0 +
stack size (kbytes, -s) 10240 +
cpu time (seconds, -t) unlimited +
max user processes (-u) 1024 +
virtual memory (kbytes, -v) unlimited +
file locks (-x) unlimited
```

```
CentOS' %PQD ÜÝf ßulimit Ðu02 ßulimit Ðn0D i DŠ | • 2 <=ØÛ Ä• ' 3O~ • t 9. Ä
ï ^` aL ) m\QD &) *% Ä7#$ u bÚ: profile (.profile, .bash_profile)5 ulimit ^ €ÜÝL
¬r * - í , ¬E 3O~ • t 9 (CentOS %P).
```

OS Øyî Ž 3O -ï DŠ | • > <=ØÙ À• (CentOS %P)

```
$ cat $HOME/.bash_profile*
.. (^w)*
ulimit -u 8192*
ulimit -n 8192*
```

```
' 9 30 6QD$ /etc/security/limits.conf (CentOS %P) ØÙL ¢f # ì DŠ | å 6• (nproc)2
<=ØÙ å 6• (nofile)' 30. 9.
```

0S Øvî Ž 3O -i DŠ | • > <=ØÙ À• (CentOS %P)

```
$ cat /etc/security/limits.conf*
.. (^w)*
*      soft nproc 8192*
*      hard nproc 8192*
*      soft nofile 8192*
*      hard nofile 8192*
```

7.6. LENA O. Z^{©a} «ho¬vw

Table 25.1 %\ QD %0r * \$ ØÙ

| %& | = ^a | ‡ + O. | - ®~ « h ⁻ |) * |
|---------------------------|--|--------|------------------------------|---------------------------------------|
| ManagerO%° L D- | LENA_HOME/repository/m onitoringDB | N/A | 10MB ~ 120MB | ® |
| ManagerŒ• Ž• , í &i Uñ | LENA_HOME/repository/m onitoringDB | 7Û | N/A | H» Ý E |
| Managerí &k © | LENA_HOME/repository/m onitoringDB |) m | 1MB † * | ® |
| Managere ´ ØÛ | LENA_HOME/repository/b ackup/lena-manager |) m | 300MB † * | ® |
| Manager Server ¨¥ | LENA_HOME/repository/c ontainer |) m | 10MB / Service Cluster | Service Cluster Ä• 5 x y i & |
| ManagerDœ | LENA_HOME/logs/lena- manager | 30Û | 10MB ~ 100MB | ® |
| AgentDœ | LENA_HOME/logs/lena- agent | 30Û | N/A | H» Ý E |
| InstallerDœ | LENA_HOME/logs/lena- installer |) m | 1MB † * | ® |
| #Oe ä Dœ | #Oe ä 34^D LENA_HOME/servers/serv er_id/logs |) m | É * 5 x y i & | ^Dİ ^r = |

7.7. WAS Image OS ° j ± Š

P WAS ~ 7< : †î s \$ 9RS TA OS ' ` @. 9.

Table 26. P WAS ~ 7< †î s ` @™Š (2020° %P)

| WAS Image | OS Image |
|------------------------------------|--|
| jboss/wildfly | centos:7 |
| open-liberty:full-java8-openj9 | debian:buster (from adoptopenjdk/openjdk8-openj9) |
| store/oracle/weblogic:12.2.1.4 | Oracle Linux |
| ibmcom/websphere-traditional | ubuntu:16.04 |
| tomcat %! †î s K) tomcat:9-jdk8 | %! Image Tag\$ openJdk %! Tag FROM openjdk:8-jdk (FROM debian:buster) |

P² 3

¥ ´ µI | : LG CNS ¾ | ¤⁻ 7< ` ´ ±

¥ O\$: #² PB¾ ¬#m •³ ù ´ 10D 10 µs` †, | Øè E13 [07796]

¥ Ÿ ££: (02) 2099-6136

¥ † ¶ w : lana-support@lgcns.com