



# Apresentação e Análise da Implementação de Infraestruturas

por

Mariana Fernandes Spínola

||

Commented [FMdGQ1]: Nome dos orientadores

Junho de 2022



## **Agradecimentos**

Começo por agradecer à empresa Timestamp pela sua receção e por me dar a oportunidade de adquirir alguma experiência profissional.

Quero principalmente agradecer ao meu orientador externo, Ricardo Oliveira. Desde o início deste estágio e ao longo de quatro meses, esteve sempre disponível para me direccionar nas situações mais complexas, a explicar os conceitos e as implementações necessárias para conseguir utilizar de maneira correta os softwares referidos neste relatório, também pela sua simpatia, esforço e paciência ao longo destes meses.

Quero também agradecer ao meu orientador Filipe Quintal, pelos ensinamentos ao longo desta licenciatura, e também pela sua orientação, disponibilidade, paciência e comentários construtivos, tornando assim possível a realização, e naturalmente a transmissão do conhecimento ganho neste estágio.

Por fim, gostaria de agradecer aos meus familiares por me apoiarem e por tornarem estes quatros meses mais fáceis.

## Conteúdo

<b>Nomenclatura.....</b>	<b>6</b>
<b>1 Introdução.....</b>	<b>1</b>
1.1 Descrição do Problema .....	1
1.2 Objetivos.....	2
1.3 Estrutura do Documento .....	3
<b>2 Arquitetura .....</b>	<b>3</b>
2.1 Máquinas Virtuais.....	3
2.2 Containers/Kubernetes.....	4
2.3 Oracle Fusion Middleware.....	5
<b>3 Implementação Middleware.....</b>	<b>8</b>
3.1 Máquinas Virtuais.....	9
3.2 Containers/Kubernetes.....	12
<b>4 Conclusão .....</b>	<b>13</b>
<b>Referências .....</b>	<b>14</b>

## Lista de Figuras

FIGURA 1 - ARQUITETURA DE UMA MÁQUINA VIRTUAL .....	4
FIGURA 2 - ARQUITETURA DE UM CONTAINER .....	5
FIGURA 3 - ARQUITETURA GLOBAL DE ORACLE FUSION MIDDLEWARE.....	6
FIGURA 4 - DOMÍNIO DE UM SERVIDOR WEBLOGIC .....	7
FIGURA 5 - TOPOLOGIA DESAFIO 1.....	9
FIGURA 6 - INSTALAÇÃO JDK NO TERMINAL.....	9
FIGURA 7 - INÍCIO DA CONFIGURAÇÃO DE ORACLE FUSION MIDDLEWARE 12C INFRASTRUCTURE SOFTWARE .....	9
FIGURA 8 - FINALIZAÇÃO DA CONFIGURAÇÃO DE ORACLE FUSION MIDDLEWARE 12C INFRASTRUCTURE SOFTWARE .....	10
FIGURA 9 - FALTA AS DEPENDÊNCIAS PARA INSTALAR A BASE DE DADOS .....	10
FIGURA 10 - CRIAÇÃO NOVO UTILIZADOR PARA BASE DE DADOS .....	10
FIGURA 11 - DETALHES DA BASE DE DADOS.....	11
FIGURA 12 - LIGAR SERVIDORES DO DOMÍNIO.....	11
FIGURA 13 - INÍCIO DE SESSÃO NO DOMÍNIO .....	12
FIGURA 14 - VERIFICAÇÃO DO FUNCIONAMENTO DOS SERVIDORES DO DOMÍNIO .....	12

## Nomenclatura

VM	Virtual Machine
OSB	Oracle Service Bus
OHS	Oracle HTTP Server
RCU	Repository Creation Utility
SSL	Secure Sockets Layer

## 1 Introdução

A humanidade estará sempre a evoluir, por isso, as tecnologias têm de acompanhar esta contante mudança. Atualmente, a tecnologia potencia quase todos os aspetos de um ser humano, principalmente de um negócio. Desde as funções de um trabalhador até aos bens e serviços disponibilizados pelo dito negócio.

Uma infraestrutura são os componentes necessários para gerir ou operar certos ambientes. Estes componentes incluem o hardware, software, componentes da rede, o sistema operativo e também o armazenamento de dados, todos necessários para distribuir serviços e soluções onde as infraestruturas são necessárias. Alguns exemplos destes componentes são os computadores, os servidores e as bases de dados. Podemos por isso dizer que é basicamente todos os serviços ou tecnologias que acedemos no nosso dia-a-dia. Todos estes componentes trabalham em conjunto para que haja um processo eficiente, confidencialidade e segurança, e por isso uma comunicação eficaz entre os vários processos. Dito isto, é cada vez mais imprescindível e necessário a existência de processos mais automatizados para que, sempre que seja necessário mudar ou acrescentar algo num dos componentes, não tenhamos de modificar todos os outros processos, que este componente alterado faça algum tipo de comunicação. Esta automatização também facilitaria manutenções ou reparações de componentes, pelo que sempre que possível, devemos aliar as tecnologias uma com as outras.

Existem assim várias possibilidades para desenvolver, gerir ou apresentar aplicações variadas. Das possíveis soluções, iremos mostrar duas infraestruturas, em que é possível garantir alta disponibilidade e segurança durante a sua utilização, e posteriormente faremos uma comparação entre estas.

### 1.1 Descrição do Problema

Inicialmente, as organizações utilizavam servidores físicos que continham apenas um sistema operativo, e todos os recursos existentes nesse servidor seriam utilizados apenas para esse sistema operativo, pois não havia maneira de fazer a gestão dos recursos utilizados[1]. O problema desta situação é que poderíamos ter apenas um sistema operativo por servidor. Assim, se fosse necessário ter outro sistema operativo teríamos de

ter outro servidor físico. Então, tínhamos de arranjar uma solução para não haver tanto desperdício de recursos, e foi a partir desta situação que se originou a virtualização. A virtualização é possível quando utilizamos um hypervisor.

Um hypervisor é um software ou hardware que cria e executa máquinas virtuais. Alguns exemplos de softwares que o permitem, são por exemplo, a VM VirtualBox e VMware[2]. Pode ser instalado em qualquer tipo de computador com hardware adequado e necessário para o seu funcionamento, e com um sistema operativo.

Depois de configurada a aplicação para a máquina virtual em questão, com as bibliotecas necessárias para o seu funcionamento, este conjunto é chamado, partição.

O problema da virtualização surge quando temos várias partições para as diferentes aplicações, e queremos fazer a sua manutenção, ou até distribuição. Quando queremos criar uma máquina virtual para correr uma aplicação em específico, é necessário instalado o sistema operativo e as bibliotecas necessárias para que no final seja possível executar a aplicação. Outras desvantagens das máquinas virtuais: são compostas por uma cópia de um sistema operativo, o que leva tempo para iniciar, ser alterado, e para que fique operacional num ambiente; como temos uma cópia completa de um sistema operativo, as máquinas virtuais ocupam muito espaço no disco, e quantas mais máquinas virtuais houver, maior será a quantidade de dados duplicados para esses sistemas operativos; tendo em conta a desvantagem anterior, podemos também dizer que em termos de eficiência, as VM não são muito eficientes, pois consomem uma grande quantidade de hardware; podemos utilizar uma máquina virtual noutro hardware, mas temos de garantir que esse hardware está a utilizar o mesmo hypervisor.

Os problemas da virtualização referidos anteriormente, podem ser resolvidos com recurso à tecnologia de containers, por exemplo o Docker.

Nas secções seguintes serão apresentadas em maior detalhe, as arquiteturas, os conceitos e as principais diferenças das infraestruturas mencionadas.

## 1.2 Objetivos

Tendo em conta aos problemas e consequente solução que foram referidos anteriormente, podemos estabelecer os seguintes objetivos deste relatório:



- Compreender o que é a plataforma Oracle Middleware e qual o seu objetivo;
- Compreender o que é a plataforma Kubernetes e qual o seu objetivo;
- Criar uma infraestrutura Oracle Middleware assente em máquinas virtuais;
- Criar uma infraestrutura Oracle Middleware assente em Kubernetes;
- Identificar as diferenças entre as duas infraestruturas.

### 1.3 Estrutura do Documento

Depois de estabelecer os objetivos para relatório, iremos apresentar na secção seguinte as arquiteturas das máquinas virtuais, de containers, e também da plataforma Oracle Fusion Middleware, que será utilizada para entender as diferenças entre as implementações das outras duas arquiteturas.

Posteriormente, iremos mostrar a implementação da plataforma Middleware em máquinas virtuais e em containers.

Finalmente, iremos percorrer pelas principais diferenças entre as implementações, e consequente, a conclusão do problema introduzido.

## 2 Arquitetura

-

### 2.1 Máquinas Virtuais

Uma máquina virtual é um software que contém um hypervisor, permitindo assim a virtualização do hardware físico de uma infraestrutura, seja esta um computador ou servidor. Possibilita a abstração do hardware físico, permitindo que o hypervisor crie e corra várias máquinas virtuais numa única infraestrutura[3]. Podemos dedicar recursos específicos para as diferentes máquinas virtuais, como por exemplo, quanta memória a máquina virtual pode utilizar, os discos, e outros componentes de hardware que existam na infraestrutura.

Uma das maiores vantagens de uma máquina virtual é confiabilidade do sistema, pois como a VM está isolada da infraestrutura, se ocorrer uma falha no sistema da VM, a máquina física não é afetada[4]. Outra vantagem é que fornece uma camada de segurança,

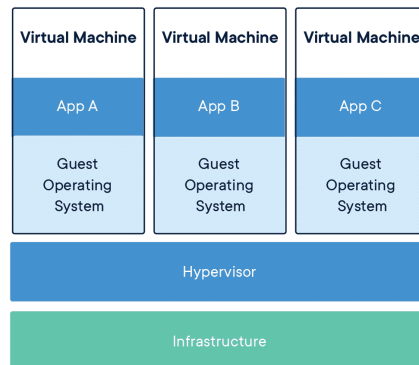
**Commented [FMdGQ2]:** todos os novos capítulos devem ficar numa nova página

**Commented [FMdGQ3]:** Este capítulo chama-se arquitetura, mas na realidade estas a fazer uma revisão de literatura, ou apresentação do estado da arte, faz falta aqui uma introdução para indicar o que é que será abordado neste capítulo.

Além disso é má prática deixar 2 títulos seguidos sem qualquer texto entre ambos

por exemplo se a VM for afetada por algum tipo de malware, não afetará a infraestrutura pois estão isoladas uma da outra.

Figura 1 - arquitetura de uma máquina virtual



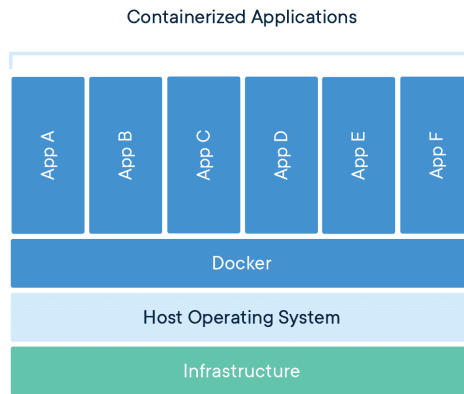
A maior desvantagem é o processo demorado de configuração e se necessário, de várias máquinas virtuais para os vários sistemas operativos ou para as diferentes configurações. Pois quantos mais processos isolados e configurações específicas são necessários, maior será a quantidade de VMs, e assim será necessário ter mais recursos do que podemos ter disponível. Outras das desvantagens são o tempo de arranque e a ocupação do disco físico, pois cada máquina virtual inclui uma cópia completa do sistema operativo, da aplicação, das bibliotecas e binários, como podemos ver na Figura 1.

Commented [FMdGQ4]: está confuso

## 2.2 Containers/Kubernetes

Um container é uma tecnologia que permite a abstração da camada da aplicação, junta o código e as dependências, e isola a aplicação com todos os ficheiros necessários para a executar[5]. Possibilita a implementação da mesma aplicação pelos diferentes ambientes sem a necessidade de alterá-la para que funcione num ambiente diferente[6].

Figura 2 - arquitetura de um container



Como podemos verificar na Figura 2, as maiores vantagens da utilização de containers são: a pouca quantidade de espaço ocupado, pois apenas temos o essencial para a aplicação funcionar; tempo de arranque é instantâneo, porque necessitarmos de menos ficheiros para conseguirmos carregar e executar uma aplicação. Outra vantagem é que também permite maior modularidade, em vez de executarmos uma aplicação complexa, em apenas um container, a aplicação pode ser dividida em módulos, como uma base de dados, o *front-end* da aplicação, etc. Ao dividirmos uma aplicação em vários módulos, podemos assim facilitar a gestão de cada módulo, e cada mudança pode ser implementada sem termos de reconstruir a aplicação por completo[7].

**Commented [FMdGQ5]:** não é boa prática colocar “etc” num documento técnico

Para melhorar e ajudar na gestão e configuração dos containers, podemos utilizar em conjunção, Kubernetes. Permitindo a facilitação da escalabilidade de uma aplicação, e principalmente a sua disponibilidade, por exemplo no caso de um container encerrar de forma inesperada, outro container com as mesmas configurações irá iniciar, garantindo que não haja tempo de inatividade[1].

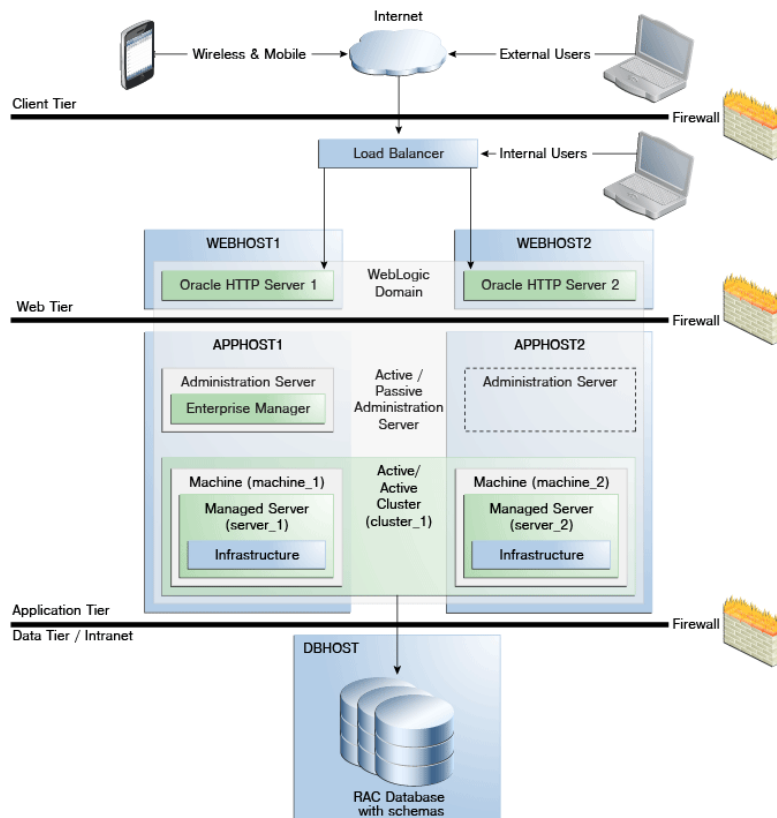
### 2.3 Oracle Fusion Middleware

Para demonstrar as diferenças entre uma máquina virtual e Kubernetes, dos componentes e instalação de uma aplicação, será utilizada a plataforma Oracle Middleware (versão 12.2.1.4).

A plataforma Oracle Middleware, desenvolvida pela Oracle, pertence a uma família com variedade de produtos que ajudam a desenvolver, implementar e distribuir, e gerir aplicações. Fornece a infraestrutura que permite a criação de aplicações, e também alta disponibilidade, quando configurada corretamente. Inclui servidores web, gestão de conteúdo dos sistemas, e outras ferramentas semelhantes que suportam o desenvolvimento de aplicações[8].

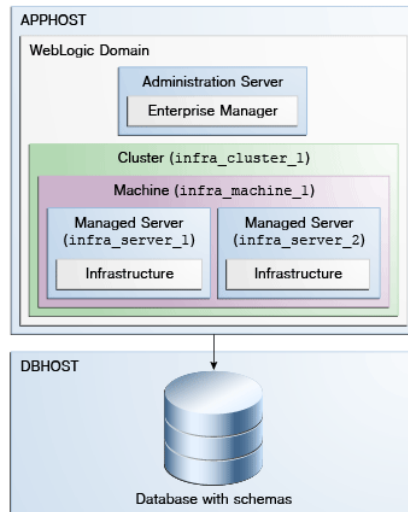
É sua função mediar entre as partes de uma aplicação, ou entre as aplicações. O design da arquitetura, Figura 3, abrange a organização, a estrutura geral, e padrões de comunicação, para aplicações e para o Middleware[9].

Figura 3 - Arquitetura global de Oracle Fusion Middleware



Agora iremos apresentar alguns conceitos e componentes que serão utilizados durante a sua implementação.

Figura 4 - Domínio de um Servidor WebLogic



Primeiro temos de compreender os principais componentes de um domínio WebLogic, presentes na Figura 4, para entendermos a implementação de Oracle Fusion Middleware.

Podemos começar pela instância principal de um domínio WebLogic chamada de Administration Server ou Admin Server. Será a partir deste componente que conseguiremos configurar e fazer a gestão de recursos para esse domínio. Também temos instâncias adicionais chamadas Managed Servers, em que todos os componentes Java, como por exemplo, as aplicações web, serão implementados nestes Managed Servers, e utilizaremos o Admin Server para fazer as configurações e sua posterior gestão.

Se agruparmos várias instâncias WebLogic conseguimos melhorar a escalabilidade e disponibilidade das aplicações. Porque num ambiente agrupado, várias máquinas irão distribuir a carga do processamento. Chamamos a este agrupamento de instâncias de cluster.

Por isso dizemos que o cluster do domínio WebLogic, consiste em um e apenas um Admin Server, e de um ou mais Managed Servers. Para que a carga seja distribuída

de forma proporcional entre os Managed Servers de um cluster, utilizamos a componente Service Bus. A vantagem de agrupar num cluster, os Managed Servers que estão num domínio Service Bus, é a possibilidade de especificação do próprio cluster como alvo de recursos de produção. O que significa que podemos aumentar a capacidade ao adicionarmos mais Managed Servers, porque normalmente colocamos como alvo de recursos o cluster, e não apenas um Managed Server sozinho[10]. Podemos então dizer, que ao utilizamos o Service Bus, estamos a melhorar a escalabilidade do domínio[11]; e também a disponibilidade, pois os componentes da aplicação são implementados em várias instâncias de servidores num cluster[12].

Um dos últimos componentes que serão utilizados é o OHS, ou Oracle HTTP Server. É um servidor web baseado na infraestrutura do servidor HTTP Apache, e inclui também, módulos adicionais desenvolvidos pela Oracle. Podemos então utilizá-lo como base para páginas estáticas, dinâmicas e aplicações web. Também pode ser usado como um servidor proxy, em que atua como um intermédio entre um cliente que solicita o recurso, e o servidor que fornece esse recurso. Alguns componentes para lidar com os pedidos do cliente:

- ouvinte HTTP, para lidar com futuros pedidos e encaminhá-los para o processamento utilitário apropriado
- módulos, para implementar e aumentar a funcionalidade básica do servidor
- Oracle WebLogic Server Proxy Plug-in, que permite que o servidor Oracle HTTP seja o *front-end* dos servidores WebLogic e de outras aplicações baseadas no Oracle Fusion Middleware

### 3 Implementação Middleware

Agora para compreender melhor as diferenças entre as máquinas virtuais e os containers, iremos demonstrar a implementação da infraestrutura Oracle Middleware nestas duas infraestruturas.

Commented [FMdGQ6]: Mesmo comentário, nova página

### 3.1 Máquinas Virtuais

Como passo inicial foi-nos indicado que deveríamos criar uma VM e criar um domínio OSB com a topologia apresentada na Figura 5.

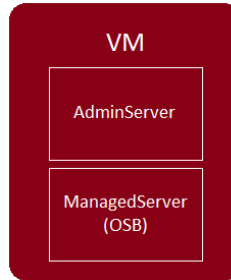


Figura 5 - topologia desafio 1

Começamos então por instalar um software que funcionasse como um hypervisor e permitisse a criação de máquinas virtuais. O utilizado para este relatório foi Oracle VM VirtualBox. Depois verificamos quais as plataformas recomendadas para instalar Oracle Fusion Middleware 12c

(12.2.1.4), foi Oracle Linux 8 (ver a documentação referenciada [13])

De seguida, iniciamos a leitura da documentação do Fusion Middleware (versão 12.2.1.4) para percebermos por onde deveríamos começar (ver documentação referenciada [14]).

1. Instalar JDK indicado para a máquina, neste caso JDK 8u311 (ver Figura 6).

Figura 6 - instalação JDK no terminal

```

[mariana@localhost Downloads]$ sudo rpm --install jdk-8u311-linux-x64.rpm
warning: jdk-8u311-linux-x64.rpm: Header V3 RSA/SHA256 Signature, key ID ec551f03: NOKEY
Unpacking JAR files...
  tools.jar...
  plugin.jar...
  javaws.jar...
  deploy.jar...
  rt.jar...
  jse.jar...
  charsets.jar...
  localizedata.jar...
[mariana@localhost Downloads]$ java -version
java version "1.8.0_311"
Java(TM) SE Runtime Environment (build 1.8.0_311-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.331-b09, mixed mode)
[mariana@localhost Downloads]$
  
```

Instalação essencial para poder utilizar os componentes e ficheiros que serão instalados ao longo desta implementação.

2. Instalar Oracle Fusion Middleware Infrastructure (ver documentação referenciada [15]).

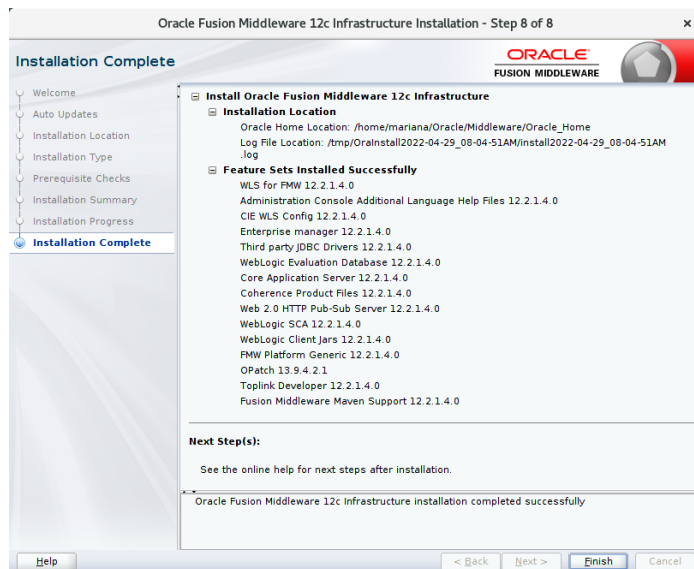
Figura 7 - início da configuração de Oracle Fusion Middleware 12c Infrastructure Software

```

[mariana@localhost Downloads]$ ls
fmw_12.2.1.4.0_infrastructure.jar  oracle-database-xe-21c-1.0-1.018.x86_64.rpm
fmw_12214_readme.html             p30188255_122140_Generic.zip
jdk-8u311-linux-x64.rpm
[mariana@localhost Downloads]$ java -jar fmw_12.2.1.4.0_infrastructure.jar
  
```

Depois de iniciarmos a configuração do software Infrastructure, temos de indicar a localização de Oracle Home, e seguimos com a instalação.

Figura 8 - finalização da configuração de Oracle Fusion Middleware 12c Infrastructure Software



3. Instalar uma base de dados para podermos configurar o domínio WebLogic, a utilizada foi Oracle Database 21c Express Edition.

Antes de podermos instalar a base de dados, foi necessário instalar as suas dependências (ver Figura 9). Voltamos a repetir o comando de instalação da base de dados depois de corrigirmos as dependências utilizando o comando:

```
sudo dnf install -y oracle-database-preinstall-21c
```

Figura 9 - falta as dependências para instalar a base de dados

```
[mariana@localhost Downloads]$ ls
fmw_12.2.1.4.0_infrastructure.jar  oracle-database-xe-21c-1.0-1.018.x86_64.rpm
fmw_12214_readme.html           p30188255_122140_Generic.zip
jdk-8u331-linux-x64.rpm
[mariana@localhost Downloads]$ sudo rpm --install oracle-database-xe-21c-1.0-1.018.x86_64.rpm
error: Failed dependencies:
        oracle-database-preinstall-21c is needed by oracle-database-xe-21c-1.0-1.x86_64
[mariana@localhost Downloads]$
```

Após a instalação da base de dados temos de configurá-la, indicando também qual a password que será utilizada para os diferentes utilizadores da base de dados, SYS, SYSTEM e PDBADMIN. Depois da configuração terminamos criamos um utilizador na máquina, neste caso foi 'oracle' (ver Figura 10).

```
[root@localhost mariana]# su - oracle
[oracle@localhost ~]$ . oraenv
ORACLE_SID = [oracle] ? XE
The Oracle base has been set to /opt/oracle
```

Figura 10 - criação novo utilizador para base de dados



#### 4. Instalar Oracle Service Bus (ver documentação referenciada [16]).

Depois de instalado, criamos as *schemas* da base de dados iniciando o RCU, Repository Creation Utility. Durante esta configuração temos de fornecer os detalhes de conexão à base de dados, que foram indicados depois da instalação da base de dados, como ver na Figura 11. Também será necessário designar um prefixo das *schemas* e seleccionar os componentes.

Figura 11 - detalhes da base de dados

```

100% complete
Database creation complete. For details check the logfiles at:
/opt/oracle/cfgtoollogs/dbca/XE.
Database Information:
Global Database Name:XE
System Identifier(SID):XE
Look at the log file "/opt/oracle/cfgtoollogs/dbca/XE/XE.log" for further details.

Connect to Oracle Database using one of the connect strings:
Pluggable database: localhost.localdomain/XEPDB1
Multitenant container database: localhost.localdomain
Use https://localhost:5500/em to access Oracle Enterprise Manager for Oracle Database XE

```

Posteriormente, foi necessário criar e configurar o domínio, usando o Configuration Wizard (seguir documentação [17]). Durante esta configuração, teremos de seleccionar Oracle Service Bus, as suas dependências serão seleccionadas automaticamente. Depois deveremos seleccionar o Domain Mode como Production, e será necessário voltar a fornecer os detalhes da base de dados e logo de seguida testar a sua ligação.

Colocamos o endereço e a porta indicados durante a configuração do domínio, e iniciamos sessão com o nome e password escolhidos, como podemos na Figura 13.

Agora apenas falta ligar o Managed Server.

Para o fazer, seleccionamos no menu do lado esquerdo, “Servers”. Depois de seleccionamos na parte superior a opção “Control”, podemos seleccionar o servidor que queremos ligar (ver Figura 12). Depois de iniciado podemos verificar o seu funcionamento (ver Figura 14).

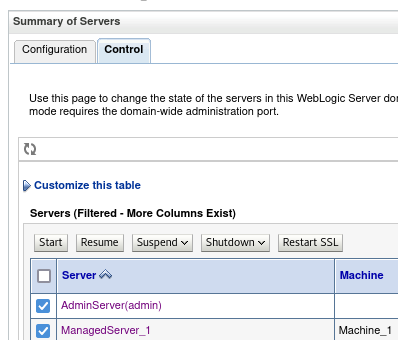


Figura 12 - ligar servidores do domínio

Figura 13 - início de sessão no domínio

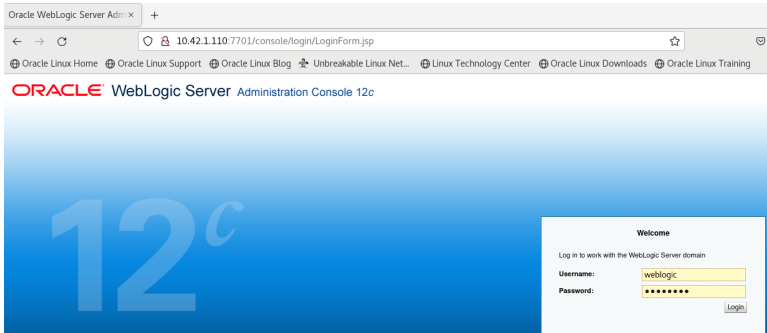
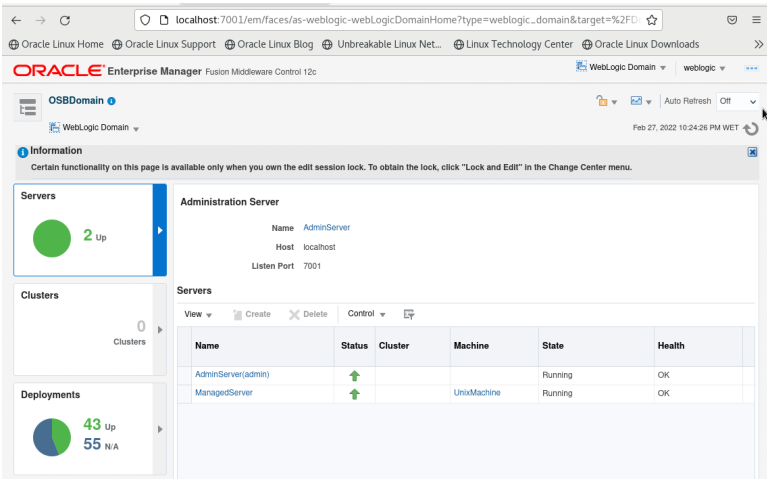


Figura 14 - verificação do funcionamento dos servidores do domínio



3.2 Containers/Kubernetes

## 4 Conclusão

## Referências

- [1] «What is Kubernetes?», *Kubernetes*.  
<https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/>.
- [2] «What is a hypervisor?» <https://www.redhat.com/en/topics/virtualization/what-is-a-hypervisor>.
- [3] «What is a Container? - Docker», 11 de novembro de 2021.  
<https://www.docker.com/resources/what-container/>.
- [4] «Container vs VM (Virtual Machine) - Know the Difference».  
<https://k21academy.com/docker-kubernetes/docker-vs-virtual-machine/>.
- [5] «Understanding containers». <https://www.redhat.com/en/topics/containers>.
- [6] «What is container orchestration?»  
<https://www.redhat.com/en/topics/containers/what-is-container-orchestration>.
- [7] «What are containers and why do you need them?», *CIO*.  
<https://www.cio.com/article/247005/what-are-containers-and-why-do-you-need-them.html>.
- [4] O. Corporation, «About Oracle Fusion Middleware», *Oracle Help Center*.  
<https://docs.oracle.com/en/middleware/fusion-middleware/12.2.1.4/ascon/overview-oracle-fusion-middleware.html#GUID-A84B6817-67D4-4E55-AA09-1856C7B0F66C>.
- [5] O. Corporation, «Understanding Middleware Architecture Design», *Oracle Help Center*. <https://docs.oracle.com/en/middleware/fusion-middleware/12.2.1.4/ascon/overview-oracle-fusion-middleware.html#GUID-9E404E02-7322-40E3-8FC3-A44516D2B30B>.
- [10] «Understanding WebLogic Server Clustering».  
[https://docs.oracle.com/cd/E23943\\_01/web.1111/e13709/overview.htm#CLUST114](https://docs.oracle.com/cd/E23943_01/web.1111/e13709/overview.htm#CLUST114).
- [11] «Understanding Oracle Service Bus Clusters».  
[https://docs.oracle.com/cd/E23943\\_01/doc.1111/e15022/cluster.htm#OSBDG158](https://docs.oracle.com/cd/E23943_01/doc.1111/e15022/cluster.htm#OSBDG158).
- [12] «Understanding WebLogic Server Clustering (12.2.1.4)», *Oracle Help Center*.  
<https://docs.oracle.com/en/middleware/fusion-middleware/weblogic-server/12.2.1.4/clust/overview.html#GUID-7082EB65-5A7B-4211-B90D-B0482DCABCA4>.
- [13] «Supported Virtualization Technologies with Oracle Fusion Middleware».  
<https://www.oracle.com/middleware/technologies/ias/oracleas-supported-virtualization.html>.

- [14] «Oracle Fusion Middleware», *Oracle Help Center*.  
<https://docs.oracle.com/en/middleware/fusion-middleware/12c-release-2.html>.
- [15] «Installing and Configuring the Oracle Fusion Middleware Infrastructure»,  
*Oracle Help Center*. <https://docs.oracle.com/en/middleware/fusion-middleware/12.2.1.4/infin/index.html>.
- [16] O. Corporation, «Configuring Oracle Service Bus Domain», *Oracle Help Center*.  
<https://docs.oracle.com/en/middleware/fusion-middleware/12.2.1.4/inosb/configuring-product-domain.html#GUID-36DFF16B-4891-46EB-9554-436A3CCF85BB>.
- [17] O. Corporation, «Configuration wizard Oracle Service Bus Domain», *Oracle Help Center*. <https://docs.oracle.com/en/middleware/fusion-middleware/12.2.1.4/inosb/configuring-product-domain.html#GUID-B6B5E795-4B47-458E-B57E-616553240460>.