

APACHE OFBIZ

In the Cloud(s)

with Deepak Dixit

Deepak Dixit



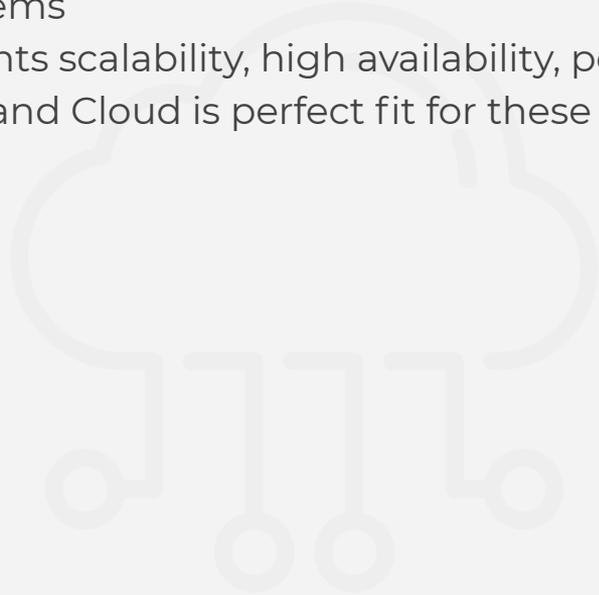
Director of Product Engineering
HotWax Commerce



Member, Apache Software Foundation
PMC, Apache OFBiz

Apache OFBiz Cloud

- Apache ofbiz is used by many companies to solve their complicated business problems
- Companies wants scalability, high availability, performance
- Apache OFBiz and Cloud is perfect fit for these companies



Agenda

1. Deployments in Cloud
2. Leverage images - AMI, Docker
3. Database options for the Cloud
4. Continuous Deployments
5. Load Balancing
6. Session management

Deployments in the Cloud

Advantage of using cloud computing

- **Cost savings**
- **Scalability**
- **Control choices**
- **Elasticity**
- **Flexibility of work practices**
- **Access to automatic updates**

Available Cloud Computing Services

- **Amazon Web Service**
- **Microsoft Azure**
- **Google Cloud Service**
- **And many more...**

OFBiz Deployment

The deployment can be made in the same way as in a standard host.

- Download Apache OFBiz
- Edit entityengine to set up the database
- Run dataload command
- Start the OFBiz server

Leverage Images – AMI, Docker

Advantage of AMI, and Dockers

- Faster set-up
- Simplified pricing
- Continuous Deployment and Testing
- Multi-Cloud Platforms
- Environment Standardization and Version Control
- Isolation
- Security

AMI - Amazon Machine Images

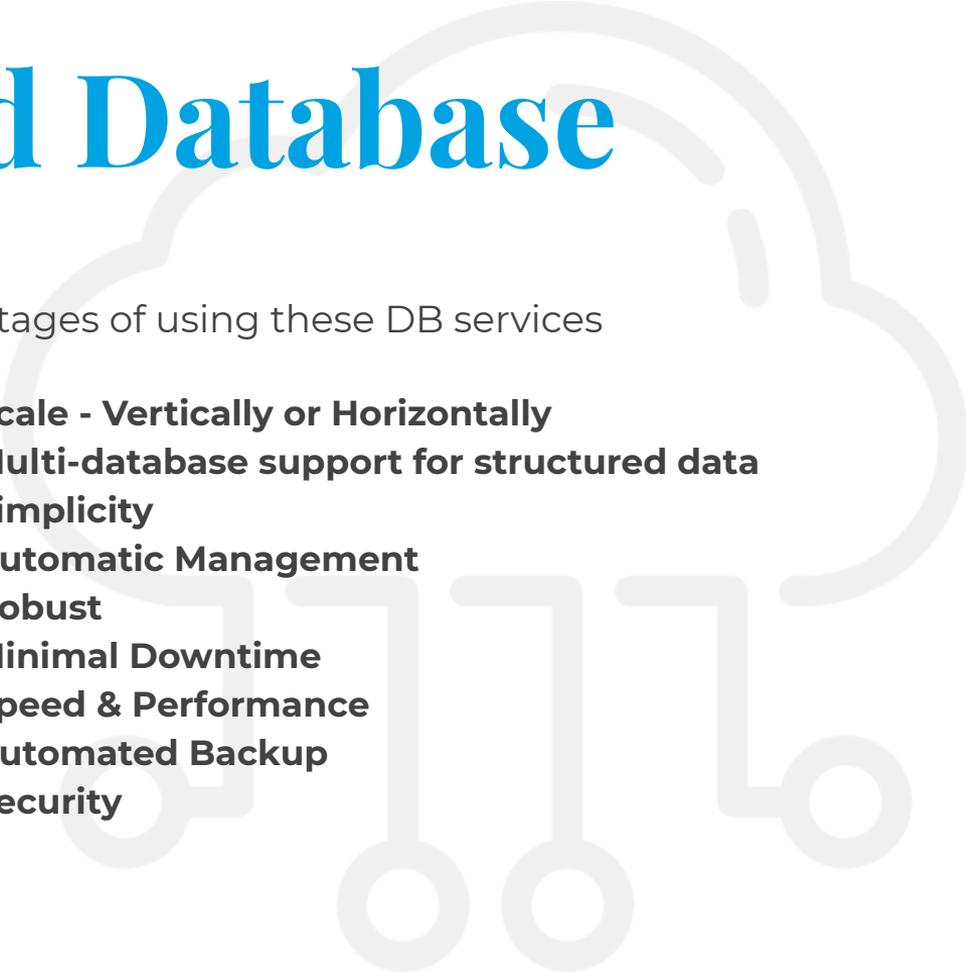
AMI provides the information required to launch an instance.

- Can launch multiple instances from a single AMI when you need multiple instances with the same configuration.
- Can use different AMIs to launch instances when you need instances with different configurations.
- 'OFBiz-for-Starter' image is available on AMI

Docker

- Create docker image
- Launch the docker container in attached or detached mode
- Access instance, browse at <https://localhost/webtools/control/main>

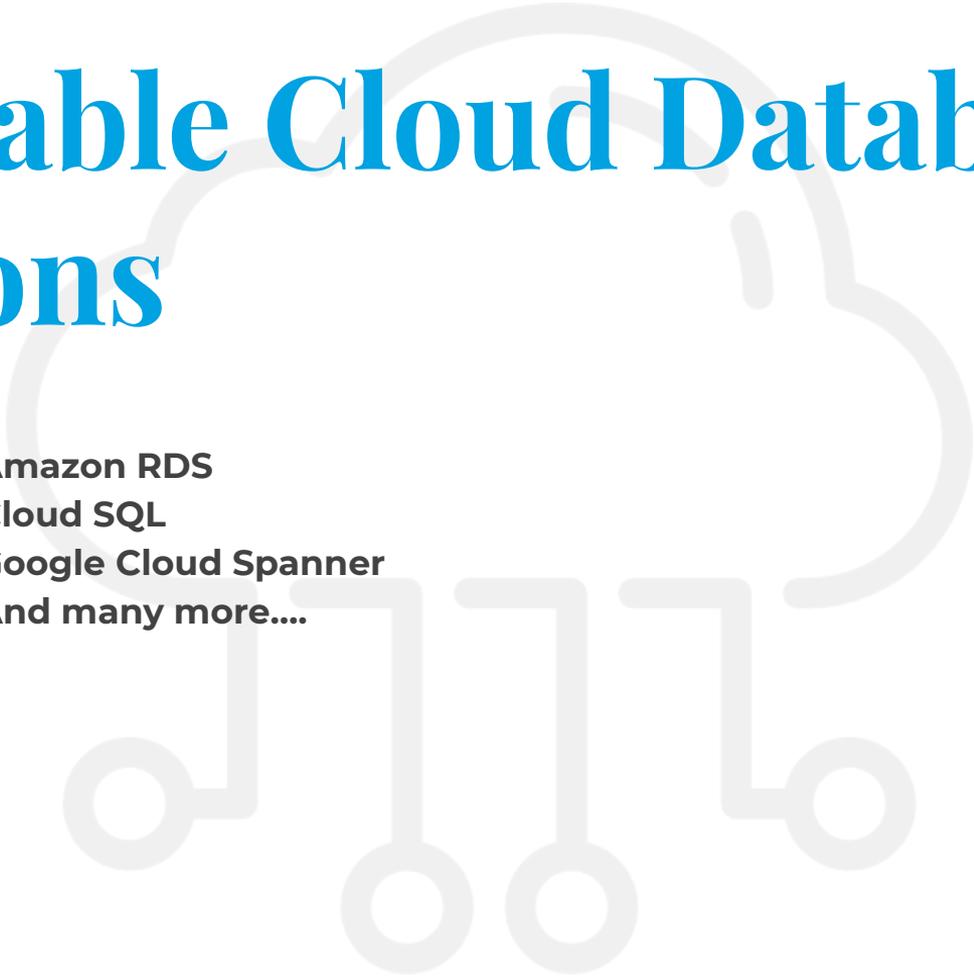
Cloud Database



Advantages of using these DB services

- **Scale - Vertically or Horizontally**
- **Multi-database support for structured data**
- **Simplicity**
- **Automatic Management**
- **Robust**
- **Minimal Downtime**
- **Speed & Performance**
- **Automated Backup**
- **Security**

Available Cloud Database Options

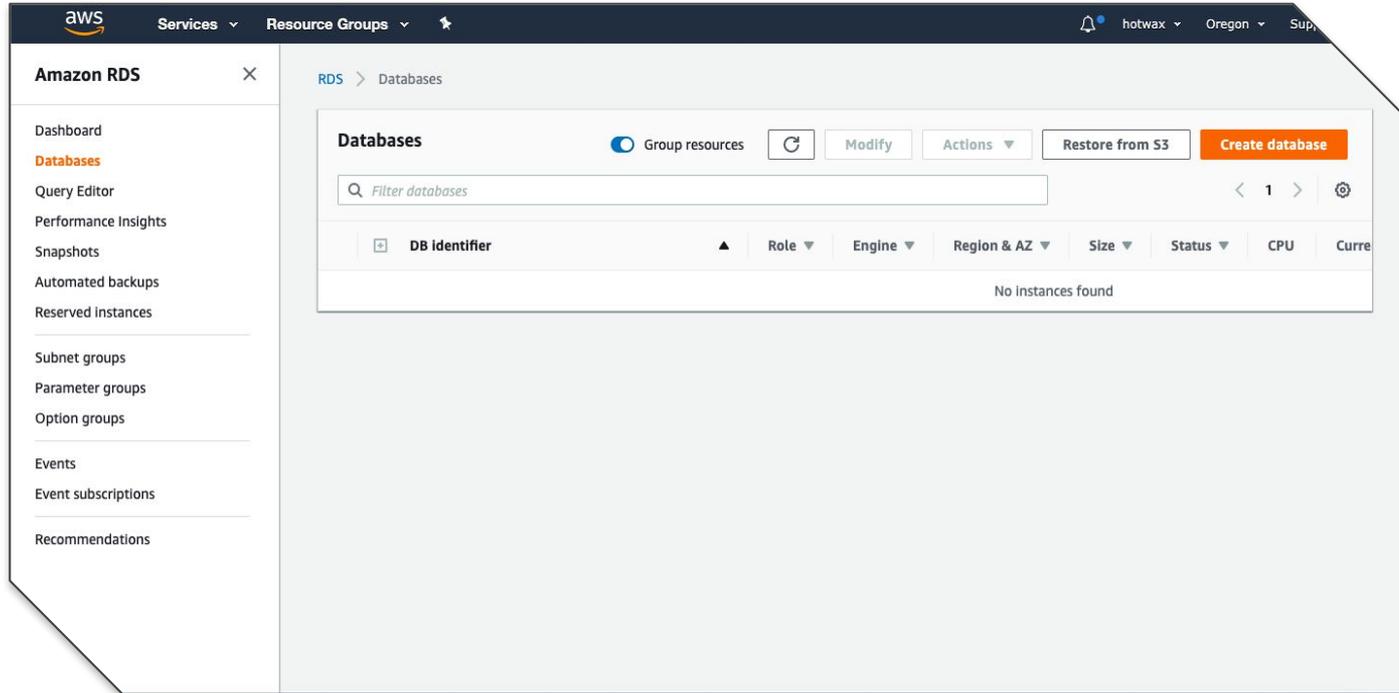
A lightbulb icon is positioned behind the title, and a circuit diagram with several nodes and connecting lines is visible in the lower half of the slide.

- Amazon RDS
- Cloud SQL
- Google Cloud Spanner
- And many more....

Configuring Amazon RDS

1. Create Database (MySQL is taken as example)
2. Add database related information
3. Refer the endpoint details
4. Update the entityengine.xml with database configuration

Create Database (MySQL is taken as example)



Create Database (MySQL is taken as example)

RDS > Create database

Create database

Choose a database creation method [info](#)

Standard Create
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

Easy Create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options

Engine type [info](#)

Amazon Aurora


MySQL


MariaDB


PostgreSQL


Oracle


Microsoft SQL Server


Add database related information

▼ Additional configuration
Database options, backup enabled, backtrack disabled, Enhanced Monitoring enabled, maintenance, CloudWatch Logs, delete protection enabled

Database options

Initial database name [Info](#)

If you do not specify a database name, Amazon RDS does not create a database.

DB parameter group [Info](#)

Option group [Info](#)

IAM db authentication [Info](#)

Enable IAM DB authentication
Manage your database user credentials through AWS IAM users and roles.

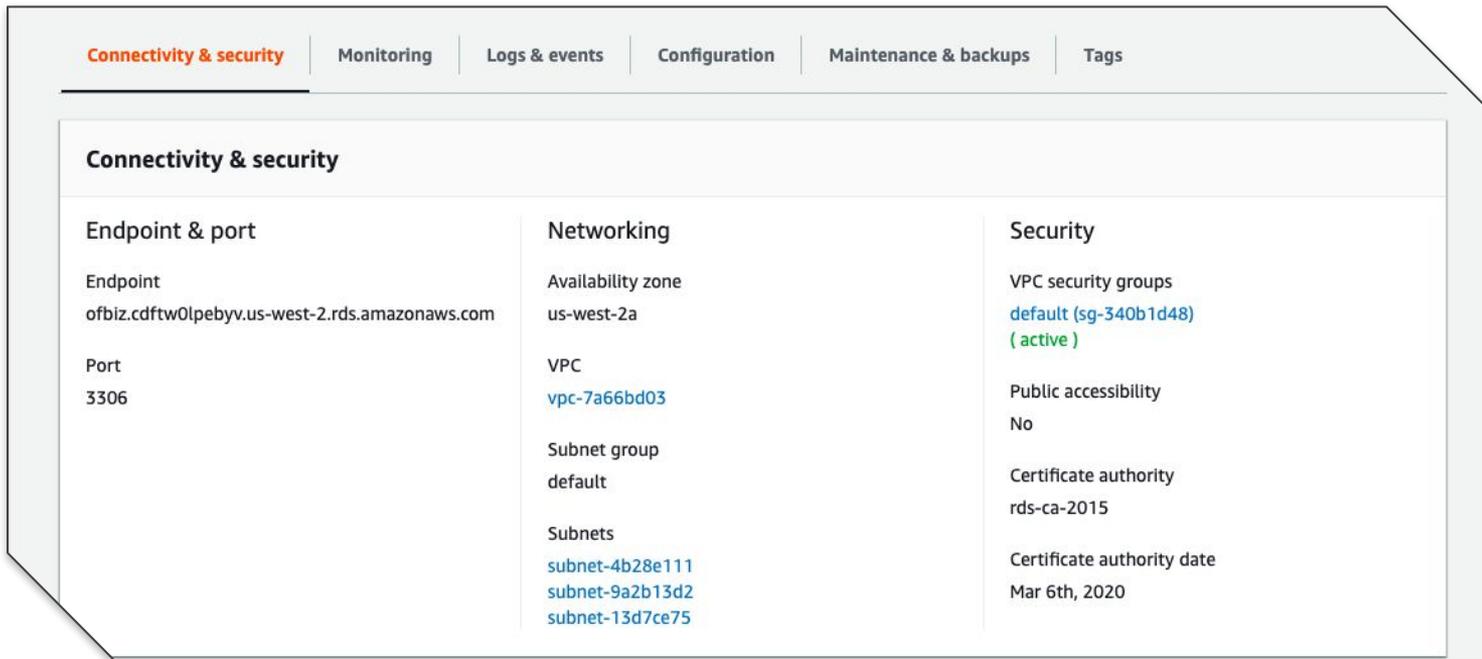
Backup

Creates a point in time snapshot of your database

Enable automatic backups
Enabling backups will automatically create backups of your database during a certain time window.

Please note that automated backups are currently supported for InnoDB storage engine only. If you

Refer the endpoint details



The screenshot displays the AWS Management Console interface for an RDS endpoint. At the top, there is a navigation bar with tabs for 'Connectivity & security', 'Monitoring', 'Logs & events', 'Configuration', 'Maintenance & backups', and 'Tags'. The 'Connectivity & security' tab is selected and highlighted. Below the navigation bar, the main content area is titled 'Connectivity & security' and is divided into three columns: 'Endpoint & port', 'Networking', and 'Security'. The 'Endpoint & port' column shows the endpoint URL and port number. The 'Networking' column lists the availability zone, VPC, subnet group, and subnets. The 'Security' column shows the VPC security groups, public accessibility status, certificate authority, and certificate authority date.

Endpoint & port	Networking	Security
Endpoint ofbiz.cdfw0lpebyv.us-west-2.rds.amazonaws.com	Availability zone us-west-2a	VPC security groups default (sg-340b1d48) (active)
Port 3306	VPC vpc-7a66bd03	Public accessibility No
	Subnet group default	Certificate authority rds-ca-2015
	Subnets subnet-4b28e111 subnet-9a2b13d2 subnet-13d7ce75	Certificate authority date Mar 6th, 2020

Update the entityengine.xml with database configuration

```
<delegator name="default" entity-model-reader="main" entity-group-reader="main"  
entity-eca-reader="main" distributed-cache-clear-enabled="false">  
  
  <group-map group-name="org.apache.ofbiz" datasource-name="localmysql"/>  
  
  <group-map group-name="org.apache.ofbiz.olap" datasource-name="localmysql"/>  
  
  <group-map group-name="org.apache.ofbiz.tenant" datasource-name="localmysql"/>  
  
</delegator>
```

Update the entityengine.xml with database configuration...

```
<datasource name="localmysql".....
```

```
.....
```

```
<inline-jdbc
```

```
jdbc-driver="com.mysql.jdbc.Driver"
```

```
jdbc-uri="jdbc:mysql://ofbiz.cdftw0lpebyv.us-west-2.rds.ama  
zonaws.com/ofbiz?autoReconnect=true&characterEnc  
oding=UTF-8"
```

```
jdbc-username="XXXXX"
```

```
jdbc-password="XXXXX"
```

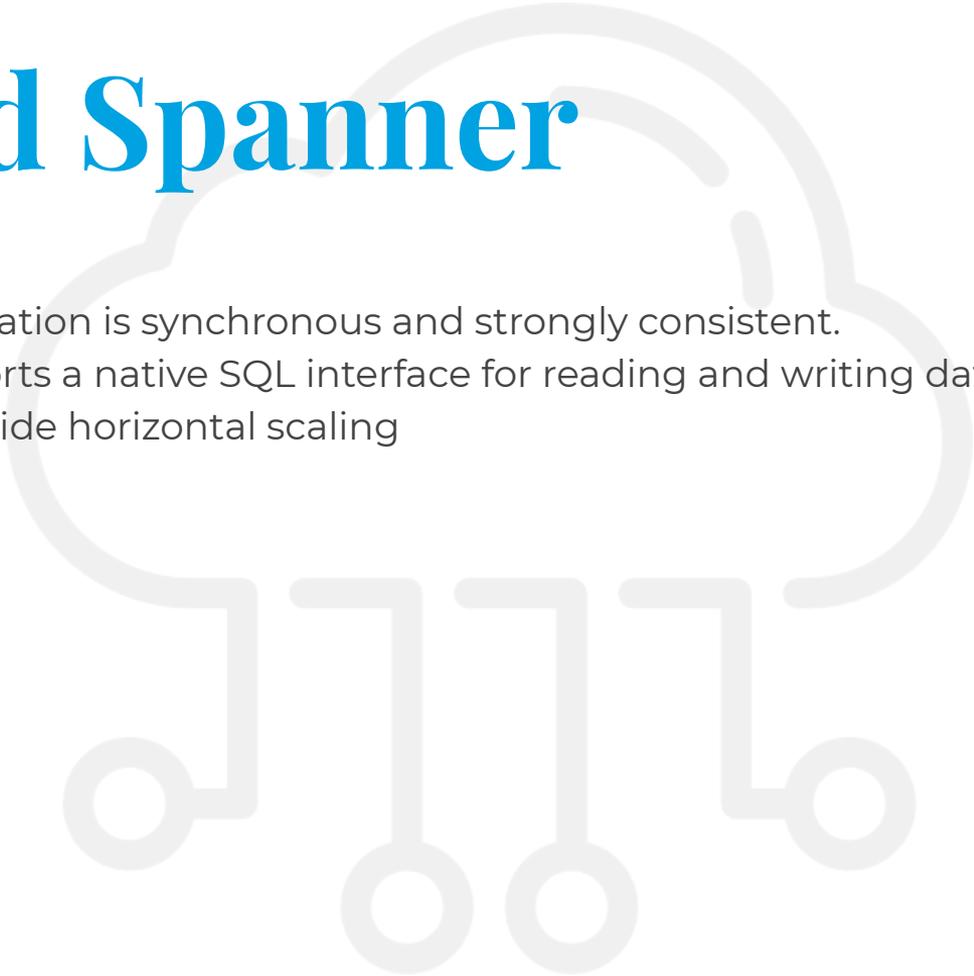
```
isolation-level="ReadCommitted"
```

```
pool-minsize="2"
```

```
pool-maxsize="250"
```

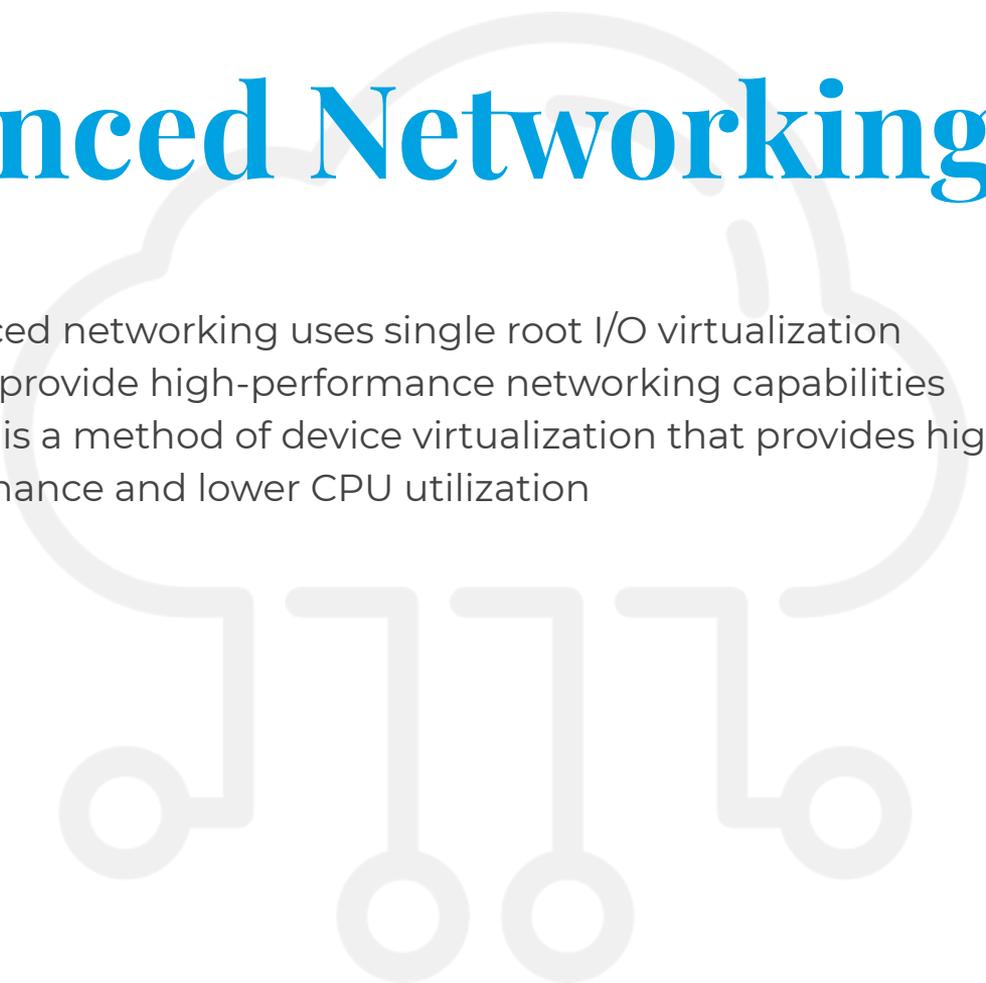
```
time-between-eviction-runs-millis="600000"/>
```

Cloud Spanner



- Replication is synchronous and strongly consistent.
- Supports a native SQL interface for reading and writing data.
- It provide horizontal scaling

Enhanced Networking



- Enhanced networking uses single root I/O virtualization
- It used provide high-performance networking capabilities
- SR-IOV is a method of device virtualization that provides higher I/O performance and lower CPU utilization

Continuous Deployments



- Automate the repetitive tasks and focus on actual testing
- Make deployments frictionless without compromising security
- Scale from a single application to an Enterprise IT portfolio
- Create workflows across the development, testing, and production environments

Advantage of using CI/CD in OFBiz

- **Faster Software Builds**
- **Time-to-Market**
- **Improvements to Code Quality**
- **Efficient Developers**

GitLab CI/CD example

Setting up GitLab Runner

- Install GitLab Runner on machine
- Registering Runners
- Assigning Runners to a Project
- Set Up SSH Key for GitLab Runner
- Enable ssh permission to instance user
- Configure .gitlab-ci.yml file

Load Balancing

Load balancing refers to efficiently distributing incoming network traffic across a group of backend servers. This ensures no single server bears too much demand

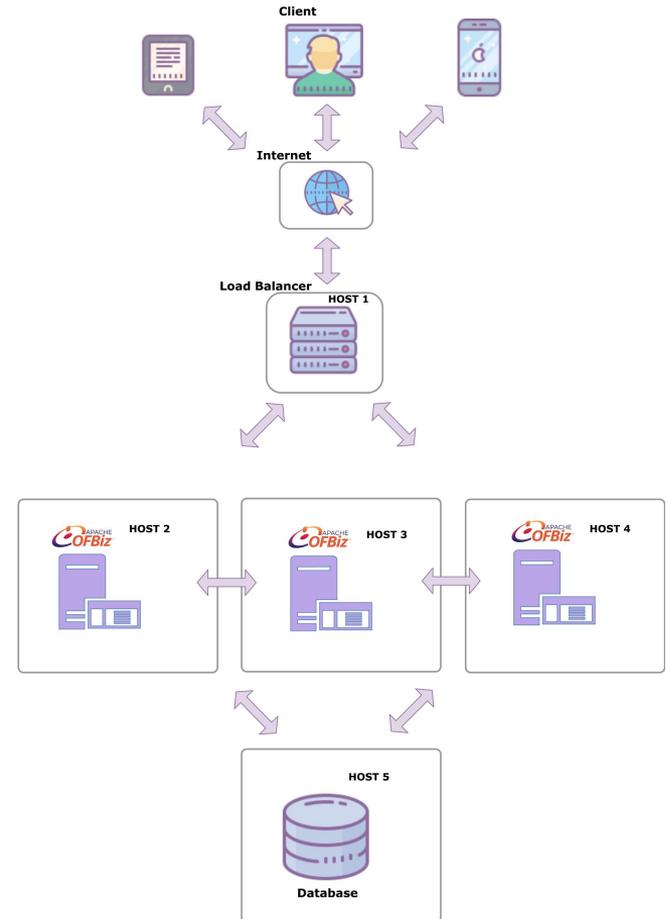
Advantage of Load Balancing

- **Increased Scalability**
- **Redundancy**
- **Reduced Downtime, Increased Performance**
- **Efficiently Manages Failures**
- **Increased Flexibility**
- **High availability**

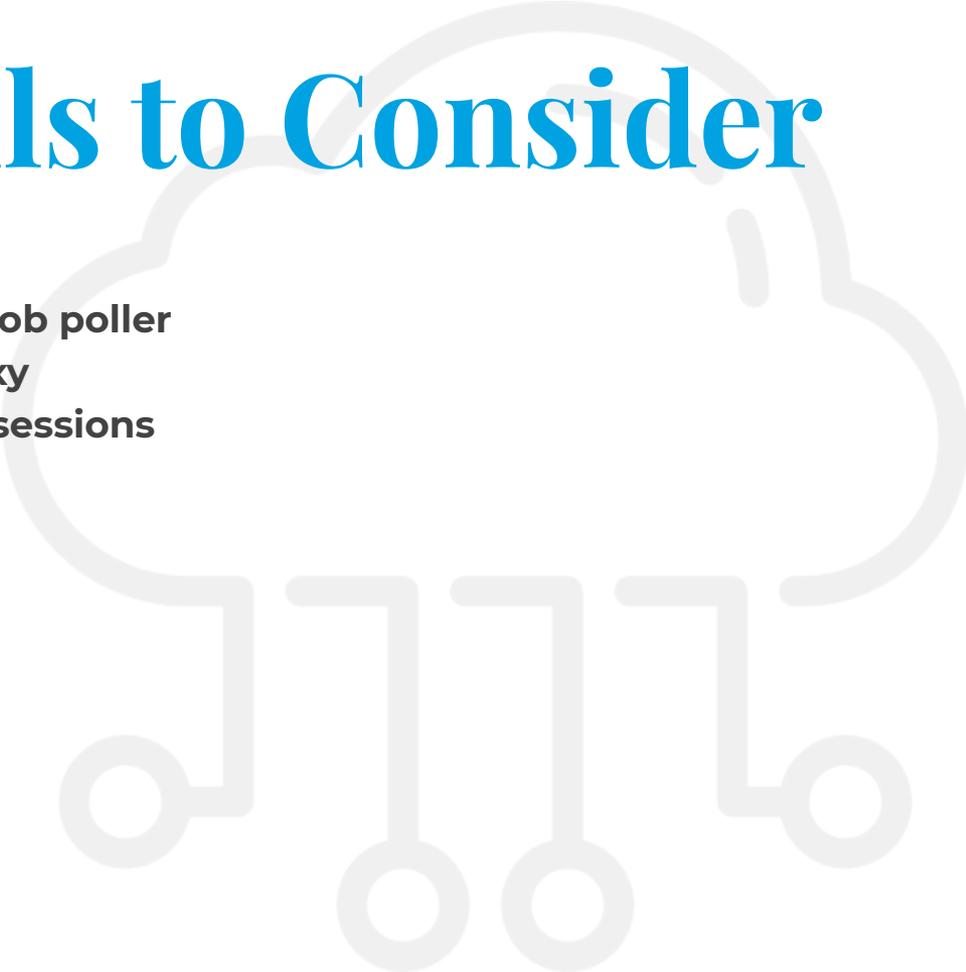
Introduction to Load Balancing an OFBiz Cluster

Load Balancing Architecture

- Setup multiple OFBiz host
- OFBiz host with single database
- Setup Load Balancer



Details to Consider



- **OFBiz job poller**
- **HAProxy**
- **Sticky sessions**

Session management



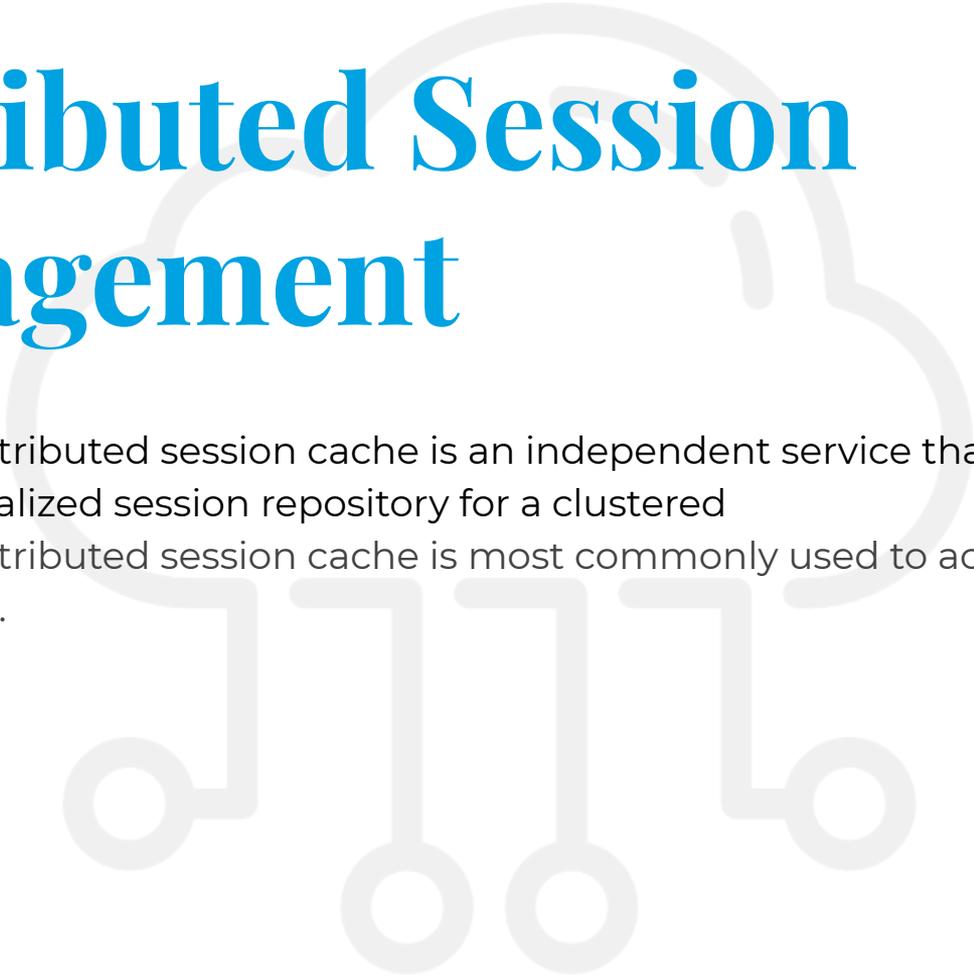
- Session management is the process of securing multiple requests to a service from the same user
- It help to achieve high performance

Sticky Session Management



- Sticky sessions is most commonly maintained by the session's `jvmRoute`
- When using sticky sessions, servers within your network don't need to exchange session data
- Sticky sessions allow for more effective utilization of your application's RAM cache, resulting in better responsiveness.

Distributed Session Management



- The distributed session cache is an independent service that acts as a centralized session repository for a clustered
- The distributed session cache is most commonly used to achieve failover.

Recap and References

- Docker / AMI images
- OFBiz deployment on Cloud
- Cloud services
- Cloud database
- CI/CD
- Session Management
- <https://docs.gitlab.com/ce/ci/>
- <https://aws.amazon.com/rds>
- <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/enhanced-networking.html>
- <https://www.hotwaxsystems.com/ofbiz/ofbiz-development/apache-ofbiz-performance/>
- <https://www.hotwaxsystems.com/quality-assurance/restore-pre-build-amazon-machine-images-to-load-test-ofbiz/>