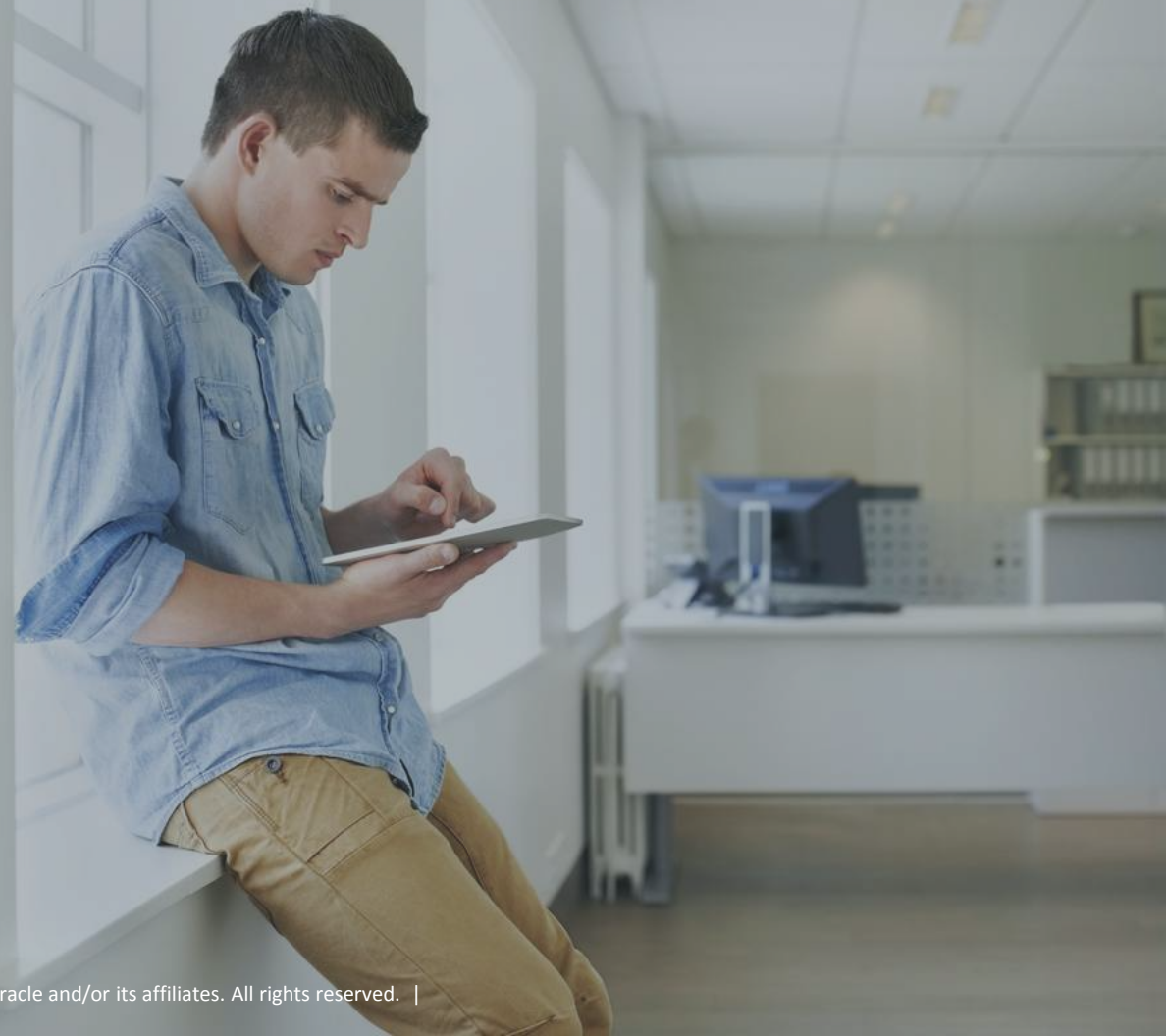
Oracle Container Cloud Service

2017. március 28.

Sárecz Lajos
Cloud Platform Sales Consultant



Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Introducing Oracle Container Cloud Service!

What is It?

- Hosted container service allows customers to deploy and run their own Docker containers
- Management and controls to orchestrate container placement and policies

What Problems Does it Solve?

- Building & maintaining Docker environments & management is complex, costly and time consuming
- Lack of example container applications and best practices to get started
- Enterprises need Docker management from dev/test through to production

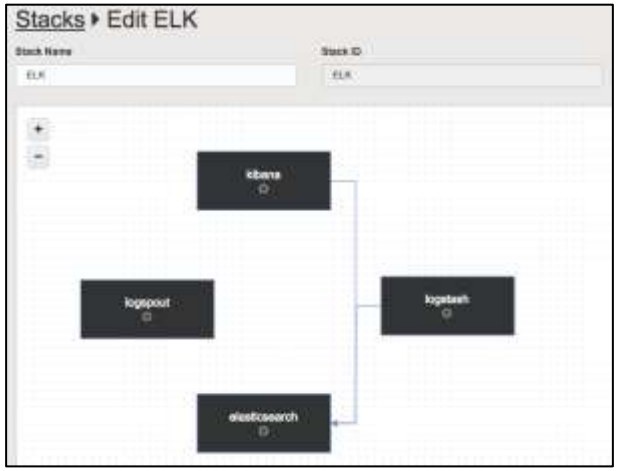
Key Benefits

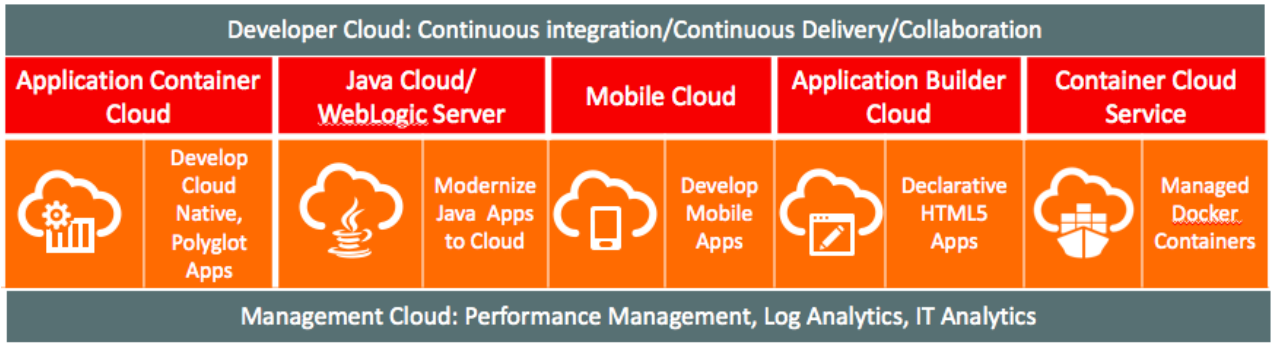
- Enables developers to get started and deploy containers quickly, DevOps teams with Docker management, visibility and control.
- Integrates with Continuous Integration & Deployment Pipelines to automate new releases.
- Supports polyglot development, open source software and microservice architectures

ORACLE

Container Cloud Service Differentiation

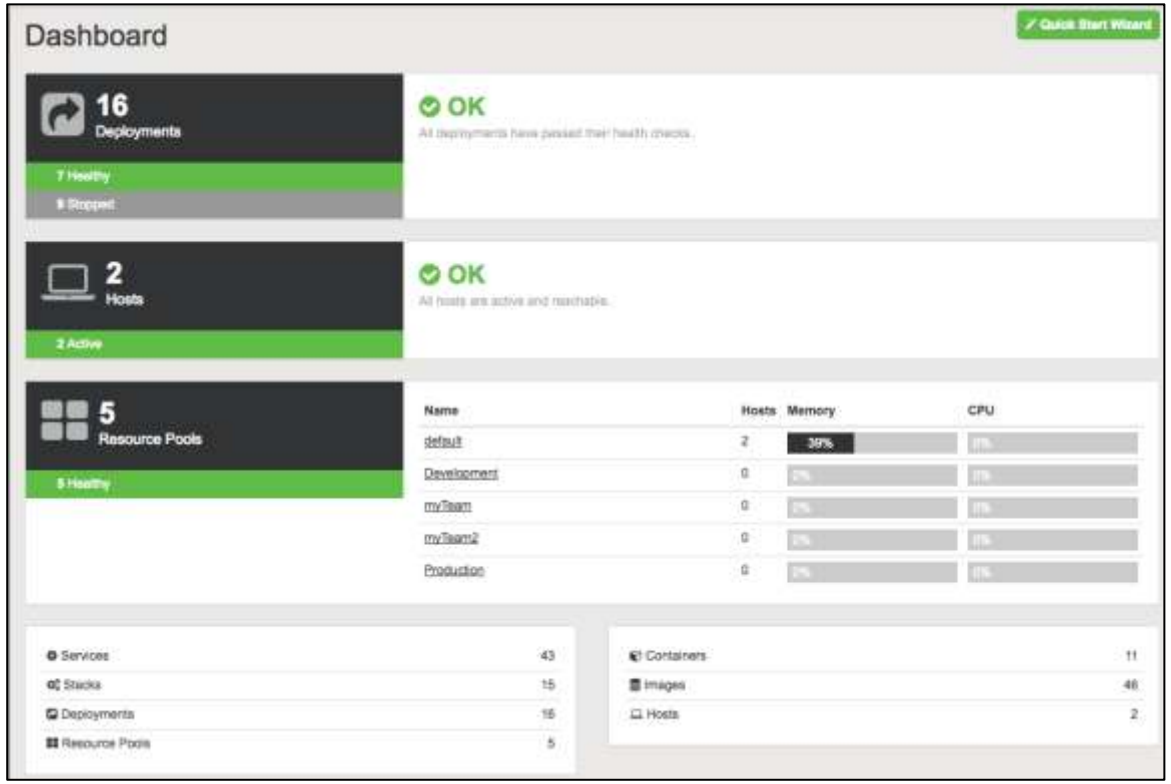
1. Easy Onboarding and Use
2. Example Application Stacks
3. Part of Rich App Dev Portfolio
4. Planned Container Hybrid Cloud with Cloud @ Customer





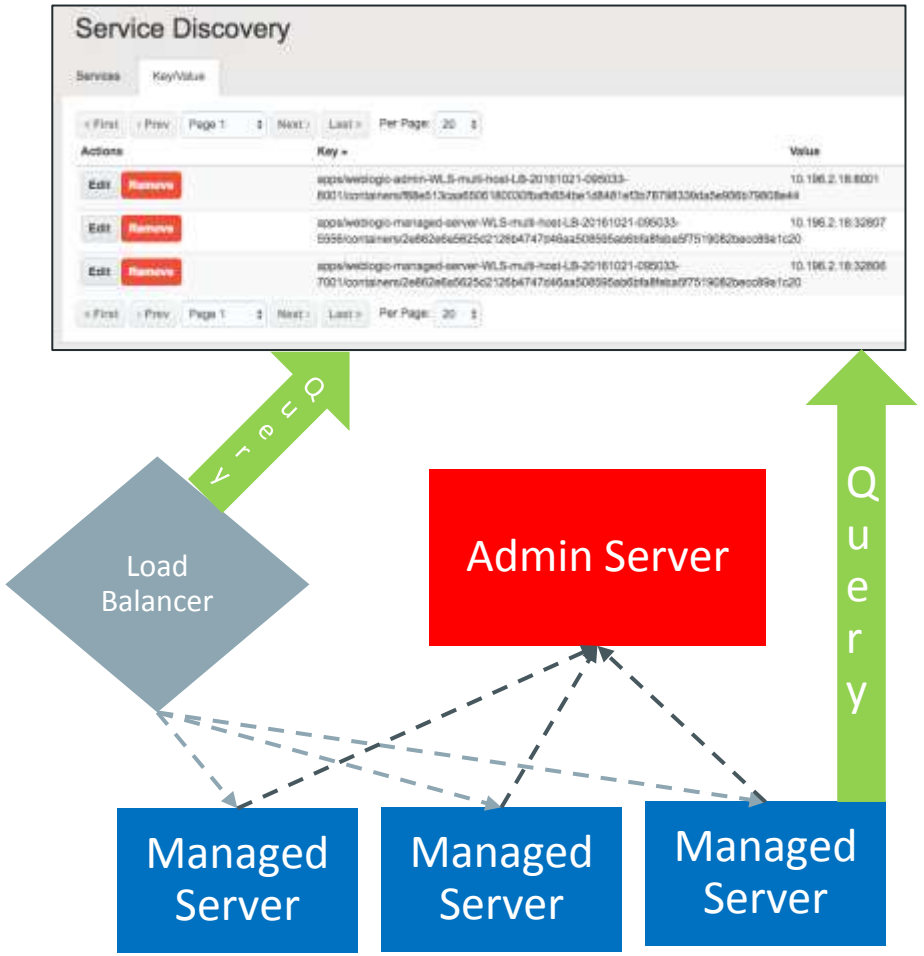
1. Easy Onboarding and Use

- Intuitive complete UI for Developers and DevOps teams
- Quickstart Wizards for rapid setup
- Dashboards give at a glance operational views
- Docker Compatible (Docker Engine and tooling like Docker Compose)



2. Example Application Stacks

- Example “Stacks” enable quick customer ramp without complex orchestration
- Service Discovery Built-In to enable multi-host app deployments
- Complete functioning examples:
  - Load Balancing
  - Logging
  - Monitoring
  - Open Source (WordPress, Redis, others)
  - WebLogic Multi-Host



Multi-Host WebLogic Stack with Load balancer

3. Part of Rich App Dev Portfolio

Unique in Blending Traditional, Cloud Native and Low Code with End to End PaaS

Oracle Platform for Cloud Application Development

BACK-END SERVICES

- Application Container
- Java
- Functions
- Mobile

FRONT-END TOOLING

- SDKs (3rd Party, JET)
- App Builder

COMMON SERVICES

- Developer Cloud (CICD)
- Container Cloud (Docker, Orchestration)
- APM/Log Analytics
- API Management
- Identity

DATA SERVICES

- Database
- MySQL
- NoSQL

INTEGRATION SERVICES

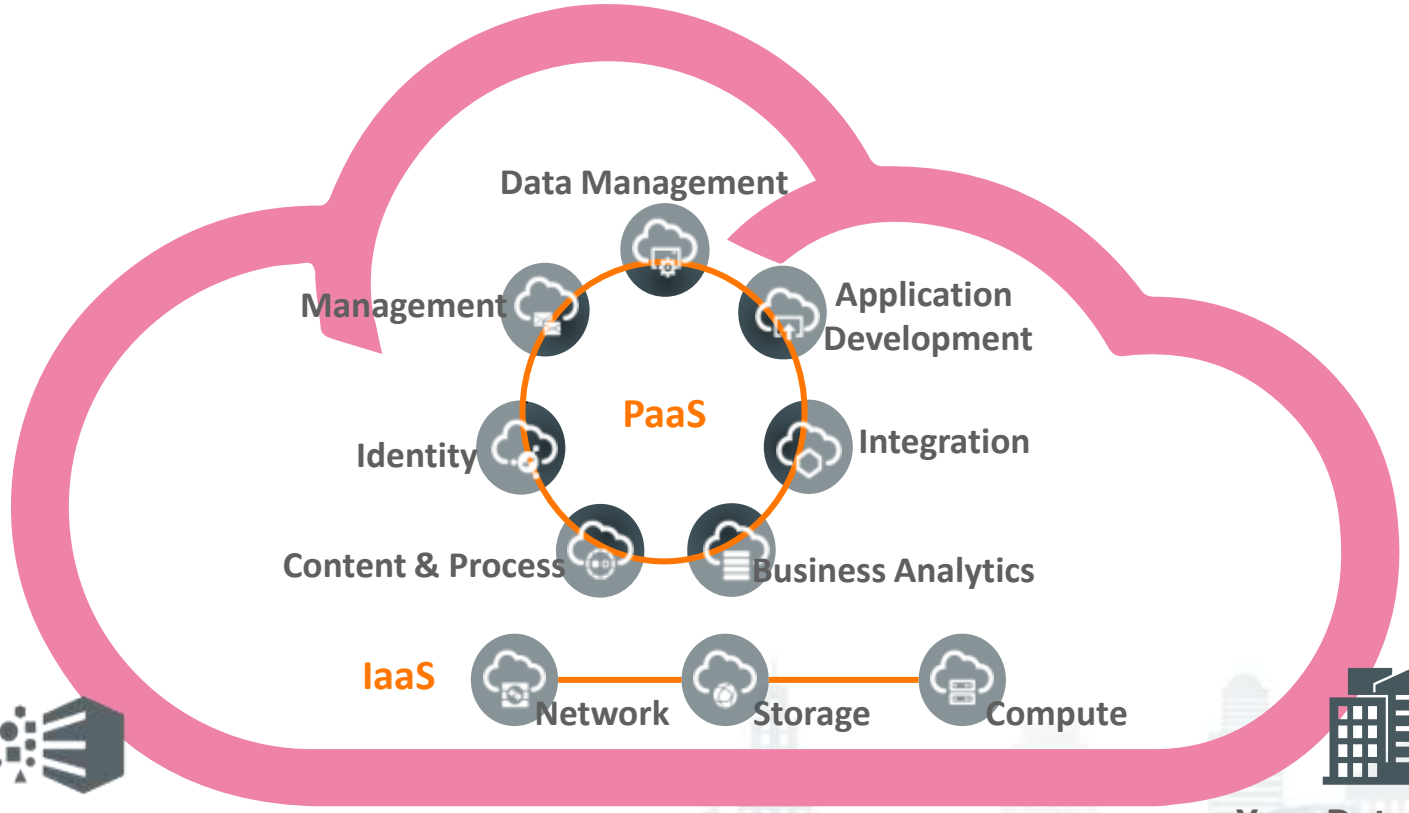
- Integration
- SOA
- Internet of Things

Compute (VM, Bare Metal) | Network | Storage

Infrastructure as a Service

ORACLE

4. Planned* Container Hybrid Cloud with Cloud @ Customer



Oracle Data Center
Oracle Cloud

Your Data Center
Oracle Cloud Machine

- Same IaaS and PaaS software
- Same updates as Oracle Cloud
- Same subscription and pay-as-you-go pricing
- Single vendor for the entire solution

*Roadmap – Safe Harbor Applies

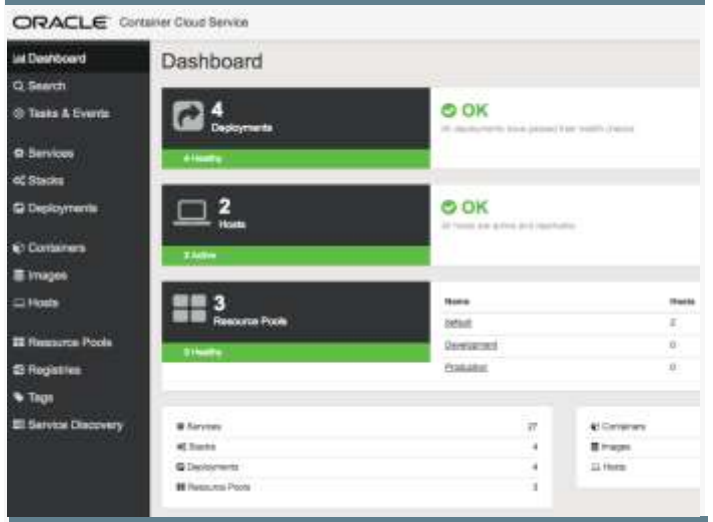
Oracle Cloud and Docker Containers

Compute CS

DIY Container Management

IaaS

Container CS



Oracle Managed Container Service

CaaS

Application Container CS

Docker-based Cloud Polyglot Platform

PaaS

Container Cloud PoC @Magyar Telekom

- The input for the excersise
  - The compute node is available at 141.145.40.244
  - Port forward is created to access the repos:
    - Configuration Repo (git):
      - The username/password is occspoc/********
      - It is on the node's port 8443
    - Artifact Repo
      - The username/password is occspoc/********
      - It is on the node's port 8444
  - Image name is energy.deconv.deconv-docker:2016.11.21-15.55.42

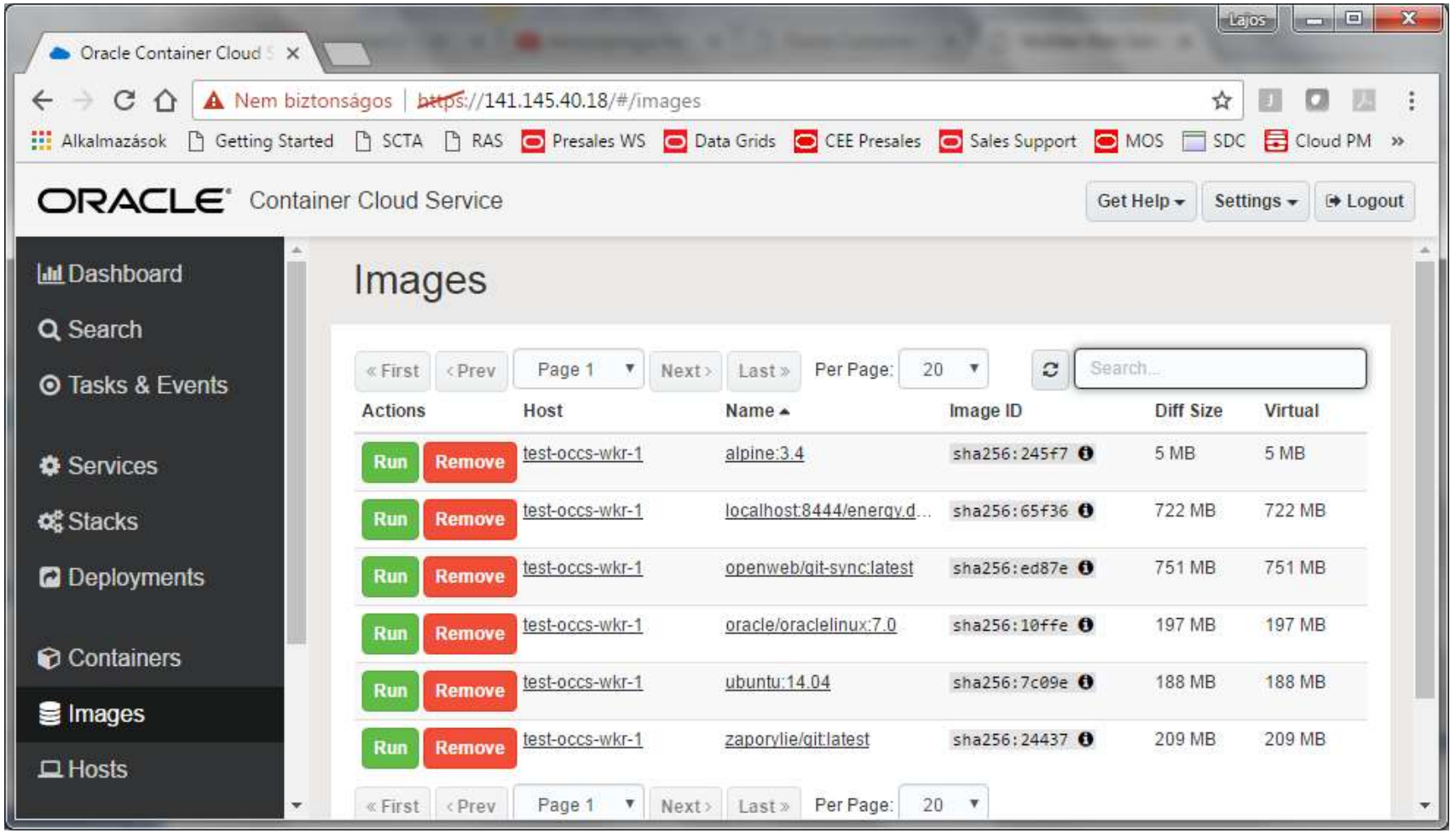
First steps in command line

```
[opc@test-occs-wkr-1 ~]$ sudo docker login  '--username=occspoc' '--password=**********' "--
email=lajos.sarecz@gmail.com" localhost:8444 WARNING: login credentials saved in /root/.docker/config.json
Login Succeeded
```

```
[opc@test-occs-wkr-1 ~]$ sudo docker pull localhost:8444/energy.deconv.deconv-docker:2016.11.21-15.55.42
2016.11.21-15.55.42: Pulling from energy.deconv.deconv-docker
10ec637c060c: Pull complete
7905d7b158eb: Pull complete
933a007dab52: Pull complete
Digest: sha256:19507220ba6b051c02c47f3032e530eb151b496551d7f9abae914b20f3721206
Status: Downloaded newer image for localhost:8444/energy.deconv.deconv-docker:2016.11.21-15.55.42
```

```
[opc@test-occs-wkr-1 ~]$ sudo docker images
REPOSITORY                                    TAG                 IMAGE ID      CREATED             SIZE
alpine                                        3.4                 245f7a86c576  6 days ago          4.803 MB
ubuntu                                        14.04               7c09e61e9035  10 days ago         187.9 MB
localhost:8444/energy.deconv.deconv-docker    2016.11.21-15.55.42 65f36ef7a663  3 months ago        721.8 MB
```

List of Images on OCCS Console UI



Access the GIT configuration repository

- Need to sync the git repo to OCCS
- Developer Cloud Service has GIT support, but we couldn’t use it
- Created a service using an image from dockerhub: https://hub.docker.com/r/openweb/git-sync/
- Set the URL for the GitLab repo: https://**occspoc:********@localhost**:8443/energy/deconv-app.git
- Set the volume path for the repo in energy.deconv.deconv-docker container
- Set --network-mode=host in the docker run command so the clone request is sent over ssh tunnel
- Set the environment variable GIT_SSL_NO_VERIFY=true