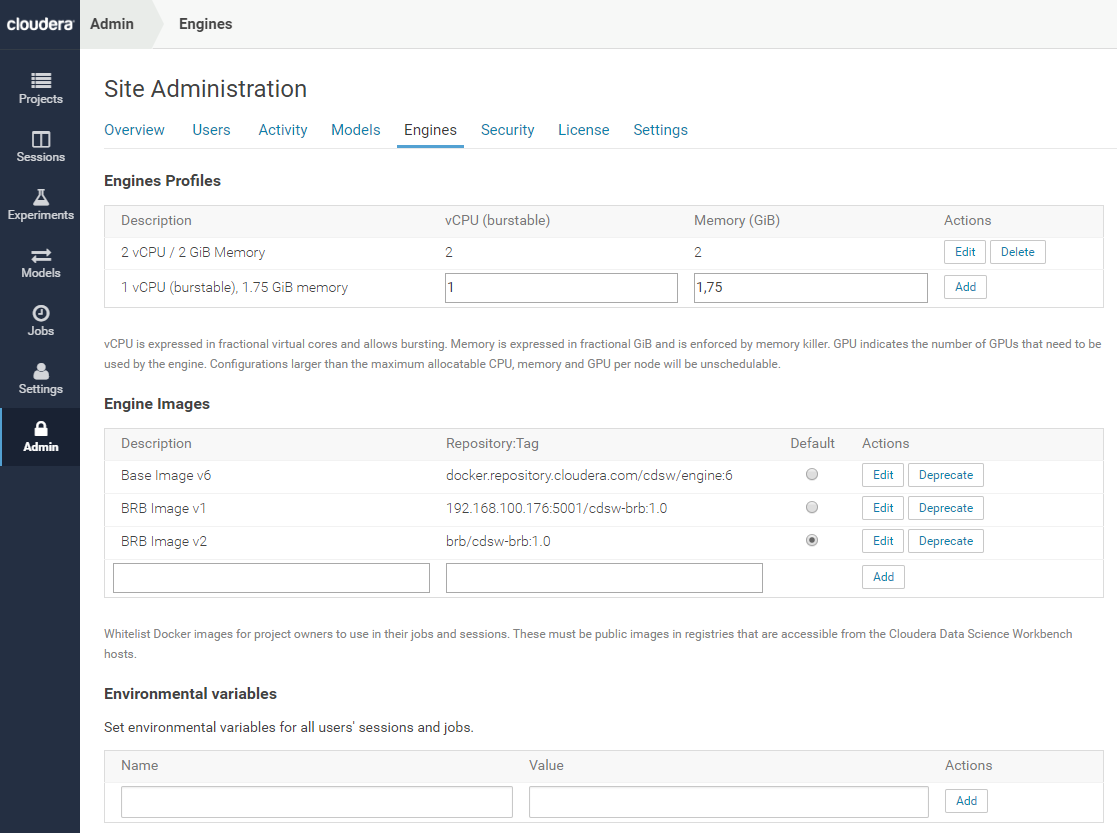
# CDSW

## Instalação

* Requisitos mínimos:
  + CDH 5.7 or higher/Cloudera Manager 5.13 or higher
  + CDH 6.1.x or higher/Cloudera Manager 6.1.x or higher
  + CDH5/6 - CDS 2.1.x
  + JDK 8
  + 100+GB – Root Volume – requerido no Master e Workers nodes
  + 1TB Application Block Device (/var/lib/cdsw) – requerido no Master
  + 500GB HDD – Docker Block Device – requerido em todos os Master e Worker nodes
  + 32+GB RAM – Master e Workers
  + 16+ CPU (vCPUs) cores – Master e Workers
* Wildcard DNS Subdomain
  + \*.cdsw.<your\_domain>.com
* Apache Spark 2
* Iptables – Desabilitar regras pré-existentes e adicionar regras default para uso com o Kubernetes
* Configuração /etc/sysctl.d/k8s.conf
* SELinux desabilitado ou modo permissivo
* CDSW não suporte servidor de DNS local
* Criar o usuário no HDFS caso ainda não tenha sido criado
  + hdfs dfs -mkdir /user/<username>
  + hdfs dfs -chown <username>:<username> /user/<username>
* Testar o Spark
  + CDH 5
    - spark2-submit --class org.apache.spark.examples.SparkPi --master yarn --deploy-mode client /opt/cloudera/parcels/SPARK2/lib/spark2/examples/jars/spark-example\*.jar 100
  + CDH 6
    - spark-submit --class org.apache.spark.examples.SparkPi --master yarn --deploy-mode client SPARK\_HOME/lib/spark-examples\*.jar 100

## Estrutura do CDSW

* Área administrativa
* Engine (Memória/CPU, Images, Variáveis de Ambiente, Pontos de Montagem)
  + Imagem de uma Docker que irá rodar na máquina Master emulando o ambiente para desenvolvimento das aplicações, modelos e experimentos
  + Mostrar tela de Engines na área administrativa
  + 
* Projeto
  + Projeto irá conter todos os arquivos de código para a aplicação bem como arquivos de dados de exemplo, arquivos de configuração, entre outros para execução do projeto
* Sessão
* Team
* Experimentos
* Modelos

### Interagindo com o CDSW – [LAB01](file:///C:\Users\jeferson.braga\Documents\Laboratorios\CDSW\lab01)

* Criar o projeto com nome de lab01
* Adicionar o arquivo README.md (liberar em local compartilhado)
* Escrever em markdown (pegar exemplo e liberar em local compartilhado)
* Fazer upload de arquivo ()
* Criar um diretório
* Fazer criar arquivo de código em Python e R
* Fazer o download de um arquivo criado
* Mover um arquivo
* Copiar um arquivo

https://www.cloudera.com/documentation/data-science-workbench/latest/topics/cdsw\_engines\_packaging.html

## Laboratório 02

1. Criando um novo Projeto – [LAB02](file:///C:\Users\jeferson.braga\Documents\Laboratorios\CDSW\lab02)

* Abrir um arquivo de python

Criando uma função de fatorial e usando a função pronta da linguagem

* + fatorial.py
* Abrir um arquivo de R

Criando uma função de fatorial e usando a função pronta da linguagem

* + fatorial.R

1. Instalando e Carregando pacotes Python e R – [LAB02](file:///C:\Users\jeferson.braga\Documents\Laboratorios\CDSW\lab02)
2. Trabalhando com datasets – [LAB02](lab02/dataset)
   1. Exemplos em Python e R
      1. Ler o dataset em formato CSV
      2. Manipular o dataset com operações de soma, multiplicação, filtragem, agregação, mínimo, máximo, media

## Laboratório 03

### Exemplificar o funcionando do Spark em Python e R

1. Spark on Python - [LAB03](lab03/Python)
2. Spark on R – [LAB03](lab03/R)

### Mostrar os [datasets](lab03/datasets) e trabalhar com eles no CDSW usando Python, R e Scala

* Leitura dos dados
* Efetuar algumas operações (min, max, sum, mean)
* Gerar alguns gráficos (ggplot)

## Experimentos Iris Project – [LAB04](lab04)

Os experimentos são funcionalidades que habilitam você para mensurar e traçar diferentes modelos de aprendizado de máquina.

Quando está se testando algoritmos de aprendizado de máquina é muito comum trocar-se variáveis que são analisadas ou transformando de diferentes formas um determinado dado ou testando um algoritmo diferente e com isso há uma série de testes que são feitos até chegar-se a um resultado aceitável, porém a cada troca de variável esse troca não fica rastreável, causando um looping sob a solução, para isso, o CDSW fornece os Experimentos.

Esta funcionalidade proporciona a habilidade de comparar diferentes versões de um modelo e ver qual obteve melhor performance de acordo com as métricas que você estabeleceu.

Com ele é possível guardar os parâmetros utilizados como entrada, as métricas de saída entre outros detalhes. Também sendo possível gerar o modelo para ser utilizado na funcionalidade de Modelos.

Exemplos:

* Regressão Linear para predição do tempo de chega de um voo
* Regressão Linear para predição do tipo flor de Iris

## Modelos – [LAB05](lab05)

Modelos no CDSW são contêineres que rodam determinados códigos em Python, R ou Scala e disponibilizam um canal de comunicação baseado em REST API.

Os modelos podem ser configurados com determinados volumes de memória e CPU. Também pode-se definir o número de réplicas que estarão em execução, ou seja, o número de contêineres ativos que podem responder as requisições.

Um modelo pode ser reconstruído alterando-se a configuração de memória, CPU e número de réplicas gerando uma nova imagem Docker.

Também é possível monitorar o número de requisições de entrada e o número de falhas de requisição. Gerar uma nova chave de acesso e deletar o modelo.

Depois que foi encontrado a equação que representa a linha, podemos usar essa equação para predizer a hora de chegada de um voo e com isso também podemos aplicar a um modelo no CDSW onde será possível efetuar consultas através de requisição via REST API.

Exemplos:

* Regressão Linear para predição do tipo flor de Iris

## Git – [LAB06](lab06)

Para melhor gerenciamento dos projetos o CDSW disponibiliza a integração com ferramentas de versionamento de código como GitHub e BitBucket.

Há duas formas de integrar com essas ferramentas, uma delas sendo o repositório público ao qual não é exigido a autenticação do usuário para acessar e manipular o repositório. A segunda forma é do repositório privado ao qual somente determinados usuários podem acessá-lo e manipulá-lo, para isso, é necessário que ser usado uma chave SSH para que o CDSW consiga conectar ao repositório.

Uma vez que o repositório foi integrado com o CDSW será possível interagir com a ferramenta de versionamento usando determinados comandos pelo terminal uma vez que não há interface visual para fazer essas ações.

* git pull
* git push
* git commit
* git add .
* git status
* git checkout
* git remove

Git de acesso público

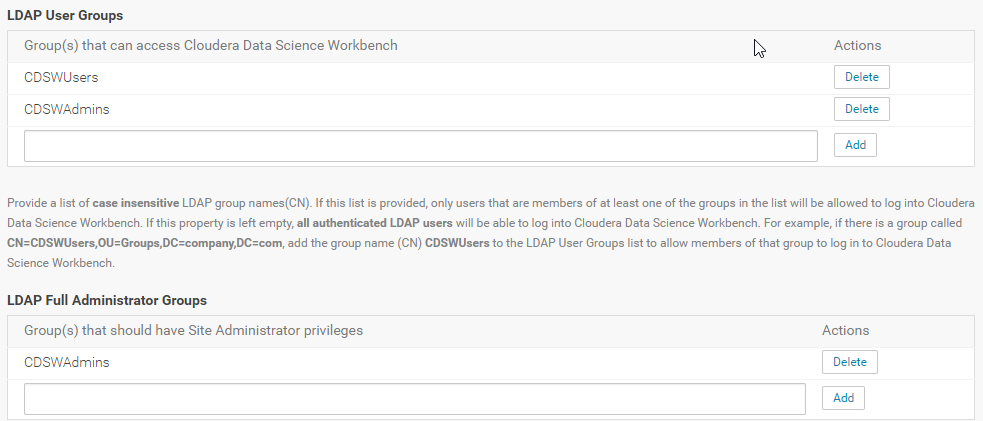
<https://github.com/cloudera/cdsw-training.git>

Git de acesso privado

* <https://bitbucket.org/>
* Seguir os passos do [README.md](lab06/README.md)

## [Integrando o CDSW com o AD](../../Manuais/CDSW%20LDAP%20integration.docx)

A integração com o *Active Directory* (AD) possibilita o melhor gerenciamento dos usuários que terão acesso a plataforma, sendo controlado o acesso através de grupos do AD.



A figura acima exemplifica dois grupos de acesso ao CDSW, sendo um o CDSWUsers de usuários comuns que não terão acesso a área administrativa, já o grupo CDSWAdmins além de permitir o acesso a criação de projetos, times e as demais operações também habilita o usuário a acessar a área administrativa da ferramenta.

## [Customizando Docker image para o CDSW](../../Manuais/Create%20a%20custom%20docker%20image%20to%20CDSW.docx)

Em determinados projetos de ciência de dados são utilizadas bibliotecas da linguagem de programação para facilitar e agilizar a manipulação destes dados, como por exemplo, *scikit-learn* que é uma biblioteca Python para aprendizado de máquina ou *ggplot* para gerar gráficos utilizando a linguagem R. Como essas bibliotecas não vem por padrão na imagem base do CDSW é necessário instalá-las a parte como já vimos anteriormente. Porém para cada projeto novo que é criado é necessário instalá-las novamente o que em determinados casos torna o processo demorado pois como essas máquinas geralmente são protegidas de acesso externo as mesmas não dispõem de acesso a internet o que impede baixar as bibliotecas. Para contornar essa situação é possível criar novas imagens de Docker para rodar dentro do CDSW. As imagens sempre devem tomar como base a imagem padrão distribuída pelo fabricante.