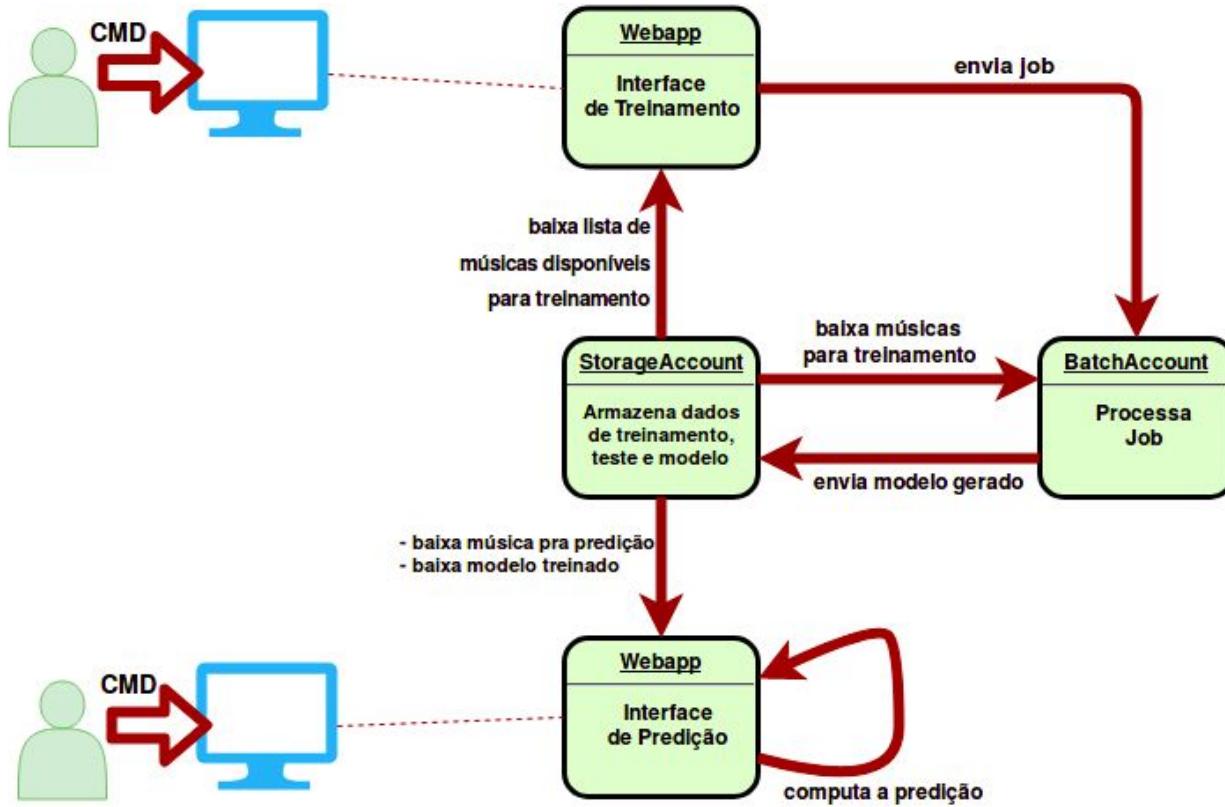




# Implementando Serviço na Azure



## Caso de Implementação:



# # O que é Webapp ?

- Azure App Service é “Platform as a Service” (PaaS);
- A aplicação roda em servidores virtualizados;
- Opções Configuráveis de permissões de acesso, load-balacing e autoscaling;
- Permite deploys automáticos a partir de repositórios Git e Docker Registries;
- Paga-se somente pelo tempo de uso apenas da máquina pedida pelo **plano de serviço**;
- <https://azure.microsoft.com/en-us/pricing/details/app-service/windows/>;

# # O que é plano do serviço de aplicativo ?

- Define um conjunto de recursos disponível para aplicações;
- **Workers** é uma unidade na escala da unidade de serviço e são esses que rodam as aplicações; Todas aplicação rodarão no mesmo worker;
- Um plano pode ter n workers, todo worker replica o mesmo trabalho;
- Problemas com volume de consumo da api, precisa-se escalar mais máquinas ;
- Problemas com demora de processamento ou falta de recurso, precisa-se de uma SKU de maior capacidade;

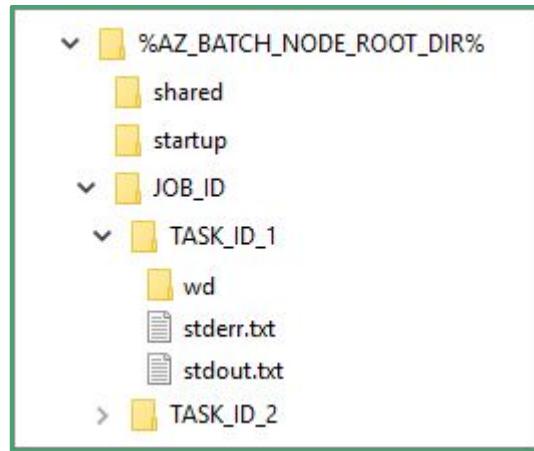


# # O que é uma conta Batch ?

- É uma forma de consumir recursos da Azure, aplicando-se workloads;
- **Pool:** coleção de nodos (VMs);
- **Job:** unidade organizacional da carga de trabalho;
- **Task:** unidade de trabalho;
- **Resource Files:** arquivos requeridos para realização da task/job.;

# # Estrutura do Nodo

- acesso SSH
- /mnt/batch/tasks;
- /mnt/batch é mapeado para dentro do container;
- As variáveis de ambiente são visíveis apenas no contexto do **task user**.
- As linhas de comando da task não rodam direto no shell, para fazer uso das variáveis, invoque o shell `/bin/sh/ \$CMD`.



## Exemplos:

AZ_BATCH_NODE_ID	The ID of the node that the task is assigned to.
AZ_BATCH_NODE_ROOT_DIR	The full path of the root of all <a href="#">Batch directories</a> on the node.
AZ_BATCH_NODE_SHARED_DIR	The full path of the <a href="#">shared directory</a> on the node. All tasks that execute on a node have read/write access to this directory.

# # Azure-CLI

- Existem outros módulos CLI para python, C++ e outras linguagens de programação;

```
$ az storage account create -n $STORAGE_NAME -g $RESOURCE_GROUP
```

```
$ az storage container create -n $CONTAINER_BLOBS_TRAIN --account-name $STORAGE_NAME --account-key $STORAGE_KEY
```

```
$ az appservice plan create -n $PLAN_NAME -g $RESOURCE_GROUP --sku B1 --is-linux -l brazilsouth
```

```
$ az webapp create -n $API_APPNAME_TRAIN -p $PLAN_NAME -g $RESOURCE_GROUP -i $API_DOCKER_TRAIN
```

# # Usando azure CLI com Json

➤ **\$ az batch job create --json-file myjob.json**

Cada instância de recurso possui uma gama extensa de parâmetros, e essas opções por vezes não estão disponíveis via argumento da função, entretanto pode-se declará-las via Json.

```
{  
  "parameters": {  
    "jobId": {  
      "type": "string",  
      "metadata": {  
        "description": "The id of Azure Batch job"  
      }  
    },  
    "poolId": {  
      "type": "string",  
      "metadata": {  
        "description": "The id of Azure Batch pool which runs the job"  
      }  
    },  
    "taskStart": {  
      "type": "int",  
      "metadata": {  
        "description": "The sweep start parameter"  
      }  
    }  
    ....  
    ....  
    ....  
  }  
}
```

# # Autenticação com Azure CLI

A recomendação principal é usar **Service Principals**:

- São identidades próprias associadas a uma conta;
- Conceder escopo necessário de permissão a ele, e usá-los nos scripts que automatizam os deploys.

```
$ az login --service-principal -u $APPLICATION_ID -p $APPLICATION_PWD --tenant $DIRECTORY_ID
```

# # Azure Active Directory - AAD

