

Tarea 1

1. Read the following article <http://martinfowler.com/articles/microservices.html> and explain what "Microservices Architecture" means 20 points.

Una arquitectura de microservices es una estructura de una aplicación, la cual consiste en dividir éste entre múltiples servicios, y cada uno corre en su propio proceso. De esta forma, cada servicio es independiente de los demás, y una modificación que se haga en uno de los servicios no obliga a que se vuelva a construir toda la aplicación.

2. Explain each line of the cc499-setup.sh file. 20 points.

echo "Start. We are about to setup the environment for your Cloud Compting class. Excited?"	Imprime en consola el texto escrito entre comillas
echo ":) :) :) :) :) :) :) :) :) :) :)"	Imprime en consola el texto escrito entre comillas
cd \$HOME	Cambia el directorio hacia HOME
mkdir Projects	Crea el directorio llamado "Projects"
mkdir .aws	Crea el directorio llamado ".aws"
cd .aws	Cambia el directorio hacia el folder ".aws"
touch credentials	Crea el archivo con el nombre "credentials"
cd ..	Se mueve un directorio hacia arriba
sudo apt-get update	Actualiza los index files de los paquetes instalados de APT (con las credenciales de superusuario)
sudo apt-get install -f	Instala los paquetes de APT que necesitan arreglarse dentro del sistema (con las credenciales de superusuario)
echo " ***** Installing git ***** "	Imprime en consola el texto escrito entre comillas
sudo apt-get install git	Instala el paquete git mediante APT (con las credenciales de superusuario)
echo " ***** Installing NodeJs ***** "	Imprime en consola el texto escrito entre comillas
sudo apt-get install nodejs	Instala el paquete nodejs mediante APT (con las credenciales de superusuario)
sudo apt-get install npm	Instala el paquete nodejs mediante APT (con las credenciales de superusuario)
sudo ln -s /usr/bin/nodejs /usr/bin/node	Crea un link simbólico entre los directorios node y nodejs, de manera que, al escribir el directorio de node, se dirige hacia el directorio de nodejs (con las credenciales de superusuario)

echo " ***** Installing AWS CLI ***** "	Imprime en consola el texto escrito entre comillas
sudo apt-get -y install python-pip	Instala el paquete python-pip mediante APT, asumiendo que todas las respuestas hacia las preguntas del sistema son "sí" (con las credenciales de superusuario)
sudo pip install --upgrade pip	Instala la versión más nueva del paquete pip mediante pip (con las credenciales de superusuario)
sudo pip install awscli --ignore-installed six	Instala el paquete awscli mediante pip, ignorando el paquete six, el cual ya se encuentra instalado en el sistema (con las credenciales de superusuario)
echo " ***** Installing yeoman ***** "	Imprime en consola el texto escrito entre comillas
sudo npm install -g yo	Instala de manera global el paquete yo mediante npm (con las credenciales de superusuario)
echo " ***** Installing grunt ***** "	Imprime en consola el texto escrito entre comillas
sudo npm install -g grunt-cli	Instala de manera global el paquete grunt-cli mediante npm (con las credenciales de superusuario)
echo " ***** Installing serverless ***** "	Imprime en consola el texto escrito entre comillas
sudo npm install -g serverless	Instala de manera global el paquete serverless mediante npm (con las credenciales de superusuario)
echo " ***** Installing dot net core ***** "	Imprime en consola el texto escrito entre comillas
sudo apt-get install curl libunwind8 gettext	Instala los paquetes curl, libunwind8 y gettext mediante APT (con las credenciales de superusuario)
curl -sSL -o dotnet.tar.gz https://go.microsoft.com/fwlink/?LinkID=809130	Transfiere los datos de la página escrita mediante SSL/TLS, y guarda la salida en el archivo dotnet.tar.gz
sudo mkdir -p /opt/dotnet && sudo tar xzf dotnet.tar.gz -C /opt/dotnet	Crea el directorio /opt/dotnet en caso de que no exista y, si se pudo llevar a cabo el comando anterior, se cambia al directorio de /opt/dotnet y se extrae el archivo dotnet.tar.gz mediante Gzip (con las credenciales de superusuario)
sudo ln -s /opt/dotnet/dotnet /usr/local/bin	Crea un link simbólico entre los directorios /opt/dotnet/dotnet y /usr/local/bin, de manera que, al escribir el directorio de dotnet, se dirige hacia el directorio de bin (con las credenciales de superusuario)

3. Create a grunt task to deploy your website to your own bucket in S3. 30 points.

Para llevar a cabo esta acción, primero se instaló grunt-aws-s3 en el Proyecto. Después, se le adjuntó al archivo Gruntfile el código “grunt.loadNpmTasks('grunt-aws-s3')”. Lo que sigue es crear un archivo que se llama “deploy-keys.json”, en donde se encuentra el siguiente código:

```
{
  "AWSAccessKeyId": "",
  "AWSSecretKey": "",
  "AWSRegion": ""
}
```

Después se agregó el siguiente código en la parte de grunt.initConfig:

```
aws_s3: {
  options: {
    accessKeyId: '<%= aws.AWSAccessKeyId %>', // Use the variables
    secretAccessKey: '<%= aws.AWSSecretKey %>', // You can also
    use env variables
    region: 'us-east-1',
    uploadConcurrency: 5, // 5 simultaneous uploads
    downloadConcurrency: 5 // 5 simultaneous downloads
  },
  production: {
    options: {
      bucket: 'favelix',
      differential: true, //Upload files that have changed
      mime: {
        'dist/assets/production/LICENCE': 'text/plain'
      }
    },
    files: [
      {expand: true, cwd: 'dist/', src: ['**'], dest: '/'},
      // CacheControl only applied to the assets folder
      // LICENCE inside that folder will have ContentType equal to
      'text/plain'
    ]
  },
  clean_production: {
    options: {
      bucket: 'favelix',
      debug: true // Doesn't actually delete but shows log
    },
    files: [
      {dest: 'app/', action: 'delete'},
      {dest: 'assets/', exclude: "**/*.tgz", action: 'delete'}, //
      will not delete the tgz
    ]
  }
}
```

```
        {dest: 'assets/large/', exclude: "**/*copy*", flipExclude:
true, action: 'delete'}, // will delete everything that has copy in
the name
    ]
  },
  download_production: {
    options: {
      bucket: 'favelix'
    },
    files: [
      {dest: 'app/', cwd: 'backup/', action: 'download'}, //
Downloads the content of app/ to backup/
      {dest: 'assets/', cwd: 'backup-assets/', exclude:
"**/*copy*", action: 'download'}, // Downloads everything which
doesn't have copy in the name
    ]
  }
},
```

Por último, se agregó el código para registrar la tarea con el nombre “deploy” con los siguientes comandos:

```
grunt.registerTask('deploy', [
  'jshint',
  'build',
  'aws_s3'
]);
```

Con esto, se puede ejecutar el siguiente código en la terminal:

```
$grunt deploy
```

4. Explain how routes work in angular and a code example. 10 points.

Para acceder a una ruta de una página web, debe de haber un archivo de javascript en la estructura de la aplicación que contenga un “route provider”. Este route provider determina la relación entre las rutas URL al correr la aplicación, y los controladores dentro de la aplicación.

Por ejemplo:

```
$routeProvider
  .when('/main', {
    templateUrl: "views/main.html",
    controller: 'MainCtrl',
    controllerAs: 'main'
  })
```

```
.otherwise({  
    redirectTo: '/main'  
});
```

En este ejemplo, se determina que, cuando el usuario escribe el nombre del URL con “/main” adjuntado, la aplicación muestra la página del archivo “main.html”, y utiliza el controlador MainCtrl, y se le adjudica el alias de “main”. Pero en caso de que el URL escrito no concuerda con ninguno de los determinados en la aplicación, la aplicación se dirige al código de “otherwise”, y en este código redirige al usuario a página “main”.

5. Explain how you can have more than one access key in your computer and how to use each one. Provide code or configuration examples. 10 points.

Teniendo un access key ya configurado en el sistema, se crea y descarga otro access key. Después, dentro del archivo ~/.aws/credentials, se modifica este de la siguiente manera:

```
[default]  
aws_access_key_id=*  
aws_secret_access_key=*
```



```
[user2]  
aws_access_key_id=*  
aws_secret_access_key=*
```

La primera sección del texto corresponde al access key del perfil “default”, mientras que la segunda sección le pertenece al perfil “user2”. En caso de que cada usuario tenga regiones y outputs distintos, estos datos se determinan en el archivo ~/.aws/config, de la siguiente manera:

```
[default]  
region=us-west-2  
output=json
```



```
[profile user2]  
region=us-east-1  
output=text
```

Para utilizar cada perfil en AWS CLI, solamente se le adjunta la opción --profile al comando que se quiera ejecutar.

```
$ aws s3 sync . s3://bucket --profile user
```

6. Explain what are you going to do different this semester based on last semester experience. 10 points.

Trabajaré más en los proyectos desde temprano, buscando terminarlos lo más pronto posible, sin menospreciar la importancia de la calidad que debe tener el producto.