

فیلد منیجر؟؟؟

Kubeflow

Spark

جلسه اول

Cloud Native Applications

متودولوژی : یک سری بست پرکتیس و راه و روش برای پیش بردن و پیاده سازی موفق یک پروژه
Cloud Native : یک سری قوانین برای پیاده سازی اپلیکیشن هایی که در محیط های مختلف قابل اجرا باشند.

12-factor : یک متودولوژی برای طراحی و پیاده سازی برنامه های Cloud Native

1. **One code base** - ما باید برای پروژه یک مخزن کد داشته باشیم که قابلیت ورژن کنترل و بازگردانی تغییرات وجود داشته باشد و بتونیم اون پروژه رو با همون کد تو محیط های مختلف اجرا کنیم، مثلا git
2. **Dependencies** - پیش نیازهای هر برنامه باید کاملا مشخص باشد و بتونه توی یک محیط ایزوله اجرا بشه.
3. **Configuration** - برنامه ها حتما و حتما باید کانفیگ هارو از طریق **متغیرهای محیطی سیستم** (environment variables) بخونه. خوندن از فایل و ... راه مناسبی نیست.
4. **Backing Service** - سرویس های دیگری که برنامه نیاز داره باید به صورتی باشد که بتونن اضافه یا حذف بشنو عملکرد برنامه مختل نشه. اتصال برنامه به برنامه ها و سرویس های دیگه از طریق لینک اون برنامه ثانویه باشد

جلسه دوم

Bash Script

اسم متغیرها توی bash به هرشکلی (حروف بزرگ، حروف کوچک، _ و اعداد) میتونن تعریف بشن به جز حالتی که اسم با عدد شروع نشه.

صدا زدن یک متغیر:

1. \$var

2. \${var}

برای پاک کردن متغیر از دستور زیر استفاده میکنیم

```
unset varName
```

```
./sc.sh var1 var2
```

```
$1 -> ./sc.sh
```

```
$2 -> var1
```

```
$3 -> var2
```

با استفاده از دستور زیر میتونیم تعداد آرگومان های ورودی رو بگیریم

```
echo $# -> number of args
```

```
echo $* -> shows all args -> creates one string include all args
```

```
echo @$ -> shows all arg -> each arg is one string, an array
```

```
echo $$ -> shows pid
```

```
echo $? -> shows exit code
```

```
echo $! -> shows a pid of last process that's running in background
```

```
echo $_ -> shows last arg that user passed
```

```
echo !$ -> shows last command in history
```

نحوه نمایش دیتای یک آرایه:

```
arrayName=(val1 val2 val3)
```

```
echo ${arrayName[0]} -> value of index 0 in array
```

```
echo ${arrayName[@]} -> all values in array
```

```
arrayName[1]=newValue
```

```
echo ${#arrayName} -> length of array
```

```
IPS=$(hostname -i)
```

```
for x in ${IPS[@]}; do
```

```
    echo $x
```

```
done
```

تو حالت بالا خط دوم خودش میاد خروجی کامند رو به آرایه تبدیل میکنه

```
IPS=$(hostname -i)
```

```
for x in "${IPS[@]}"; do
```

```
    echo $x
```

```
done
```

```
#!/bin/bash
A=10
B=20
echo `expr $A + $B`
echo $(( $A + $B ))
let C=$A+$B
```