**DESCRIPTION**

*“If Amazon was being built today it would be on AWS and it would have taken full advantage of serverless computing.”*- Andy Jassy, CEO of AWS.  
  
*“The world’s applications going forward need an app model, that is distributed,*event driven*, and serverless.”* - Satya Nadella, CEO of Microsoft at Build 2018.  
  
Serverless is an evolutionary paradigm in cloud computing where you run your applications and services without explicitly provisioning/managing the servers. Whether you are Developer, DevOps or a Product owner building your next application, you want to evaluate Serverless as a building block to host and run your application.  
  
During this meetup our guest speaker **Surya Jakhotia** (Jazz product owner, T-Mobile) will introduce you to Serverless, expand on the benefits of serverless, show you how to create serverless service in under a minute, and answer your questions.  
  
Please, bring your laptop to the meetup to try out the demo(s).

============================================

**6:30 pm** - Doors Open  
**7:00 pm** – “Create a Serverless Service in Under a Minute” by Surya Jakhotia, T-Mobile Product Owner of Jazz Serverless  
In this talk Surya Jakhotia will share why you need to evaluate Serverless as a building block to host and run your next application. The talk will also showcase how to jump-start your journey to Serverless using Jazz (https://github.com/tmobile/jazz), an open source platform built to accelerate adoption of Serverless services. We will demo how you can create a website/API/function in under a minute.  
**7:45 pm** - Q&A  
**8:00 pm** - NETWORKING/MEETUP

Global cloud security alliance

Cloud academy

*Find a way for developers to spend more time on building – Jazz was born*

*Surya Jakhotia – leads the open-source*

*Jazz in AWS accounts and contributors on GitHub*

Create auto-scaling highly available fault tolerant service

*Future of cloud applications are serverless*

Serverless is an evolutionary paradigm without explicitly provisioning/managing servers to run applications.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Physical | VM | Container | Serverless |
| Hardware | X |  |  |  |
| Host OS | X |  |  |  |
| Guest OS | X | X |  |  |
| Container | X | X | X |  |
| Application | X | x | X | X |

**Why Serverless?**

Agility

* No provision of servers
* Horizontal scaling (*supposed to handle 1 request per second but with horizontal scaling… better)*
* High availability and fault tolerance (*usually problems of engineers)*

Cost

* Never pay for idle capacity
* Pay in increments of 100ms (*good for peaks and non-peaks)*

*No such concept of under-utilization*

*FaaS (Function as a Service)*

AWS Lambda

Azure Function

Apache OpenWhisk

OpenKube

*AWS has 4-page documentation for Lambda (too long?)*

Jazz – you write the code, we do the rest

OSS product that can create serverless services

* API
* Function
* Website
* Alexa (*can build Alexa skill*)

Your instance of Jazz

* Built-in registration and authentication (*registration code*)
* Template for NodeJS, Java, Python (*best practices)*
* CI/CD *(based on Jenkins)*
* Isolated environments
* Logs
* Monitoring

*You get your own instance of Jazz*

|  |  |  |
| --- | --- | --- |
| **PoC =>** | **Pre-Prod =>** | **Prod** |
| Procure machine | Procure machine | Provision for peak and anticipated loads |
| Dependencies | Dependencies | Plan for traffic spikes |
| Patching/updates | Patching/updates | Dependencies |
| Security | Security | Patching/updates |
| Write code | Write code | Security |
| Build | Build | Write code |
| Deploy | Deploy | Build |
| - | - | Deploy |

Demo

*Jazz -* [*http://d14y3gk97btdgr.cloudfront.net/*](http://d14y3gk97btdgr.cloudfront.net/)

*CloudFront – AWS deployment in Edge*

*Provision for 8gb for dev*

*Git clone => add => commit => push*

*Generated error from Jazz. Cannot push*

*Settings =>*

*Now you can push.*

*Creating a feature branch in git repo => Jazz will pick it up, and deploy to the cloud and get different virtual environment.*

*Generates and endpoint too.*

*You can offer API service. Request can go planet-scale.*

*Jazz produces handlers for Alexa*

*Jazz allows logging*

[*http://try.tmo-jazz.net/*](http://try.tmo-jazz.net/)

Registration code: JAZZ16

T-Mobile Open Source:

<https://opensource.t-mobile.com/>

focusing on real-time alerts

Nicholas Chris? – you must work on open-source

Allie??

*Solve enterprise problem with a startup mentality*

Lorelei Newman – External VP of City U

Q&A

*With Jazz do you still need full stack developer?*

*Somehow.*

*Jazz is written and deployed in Jazz*

*AWS gives only cloud compute*

*Jazz gives best practice template codes*

*How about migration?*

*There have been apps in T-Mobile that only is in on-premise. Monolith servers.*

*You can deconstruct microservices of monolith server into small chunks.*