OCI Functions Are Fun

Simplify, Automate, and Scale with Serverless Computing

ASCEND 2025 | OATUG ORACLE APPLICATIONS & TECHNOLOGY USERS GROUP | OHUG ORACLE HCM USERS GROUP

Speaker

Mike Miller

Syntax Senior Solution Architect

- Over 25 years of working with enterprise software and information security technologies
- Experience with enterprise software implementation and support, cloud operations, and executing compliance and risk management programs.
- A CISSP, Certified Information Systems Security Professional
- Oracle ACE Associate

Syntax

- Founded in 1972 in Montreal, Canada, Syntax is a global company with 3,600+ employees across 15 countries
- Syntax provides full-stack, full-lifecycle Cloud Managed Services and Application Managed Services focused on leading ERP solutions such as JD Edwards, Oracle E-Business Suite, and SAP
- Syntax is a multicloud partner and supports OCI, AWS, Azure, GCP, and Syntax Enterprise Cloud®
- Our ERP solutions include an array of value-add services, including our AI-driven monitoring and automation platform, CxHub customer experience portal, security management, and FinOps

ORACLE | Service Partner

Expertise in
Cloud Service Solution
OCI Migration
in NAMER–North America

ORACLE | Service Partner

Expertise in
Oracle E-Business Suite
Applications to Oracle Cloud
in North America

ORACLE | Service Partner

Expertise in
JD Edwards Applications
to Oracle Cloud
in North America

ORACLE | Service Partner

Expertise in
Oracle Cloud Platform -
Oracle Cloud Platform Integration
in North America

ORACLE | Service Partner

Expertise in
CSPE: Oracle Cloud Platform -
Oracle Cloud Platform
Data Management
in North America

ORACLE | Service Partner

Expertise in
CSPE: Oracle Cloud Platform -
Oracle Database to Oracle Cloud
in North America

Agenda

1. OCI functions
   - What are they
   - Use cases and architecture
2. Creating and Deploying OCI Functions
   - Best practices
   - When things go wrong
   - Limitations
3. Live Demo
   - Execution and possibly deploy
4. Closing Comments

Show of hands:

- Already using Functions today?
- Know what requirements.txt is?
- Know what venv is?
- Dockerizing Python scripts today?
- Heard of or use OCI Cloud Shell or Editor?

Creating Functions

Overview and Troubleshooting

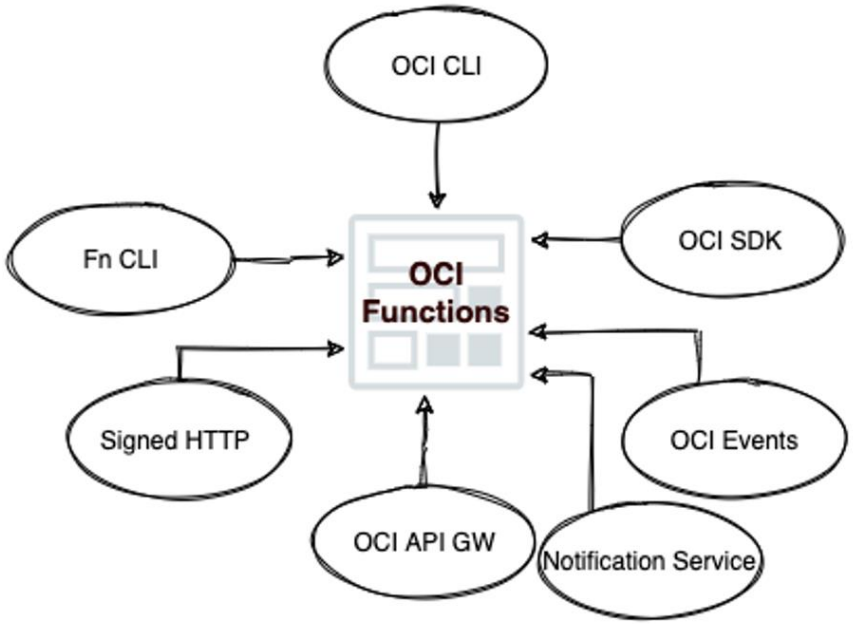
What Are OCI Functions?

Definition

- OCI Functions is a serverless compute service in Oracle Cloud Infrastructure (OCI) that lets you write and execute lightweight, scalable code without managing infrastructure.
- Based on the Fn Project, an open-source container-native serverless platform.

Key Features

- Event-Driven Execution: Functions automatically respond to events like API calls, file uploads, or scheduled tasks.
- Scalable & Cost-Effective: Scale up or down based on demand and pay only for the resources used.
- Language Support: Python, Go, Java, Node.js, Ruby, C#



Why Are OCI Functions Fun?

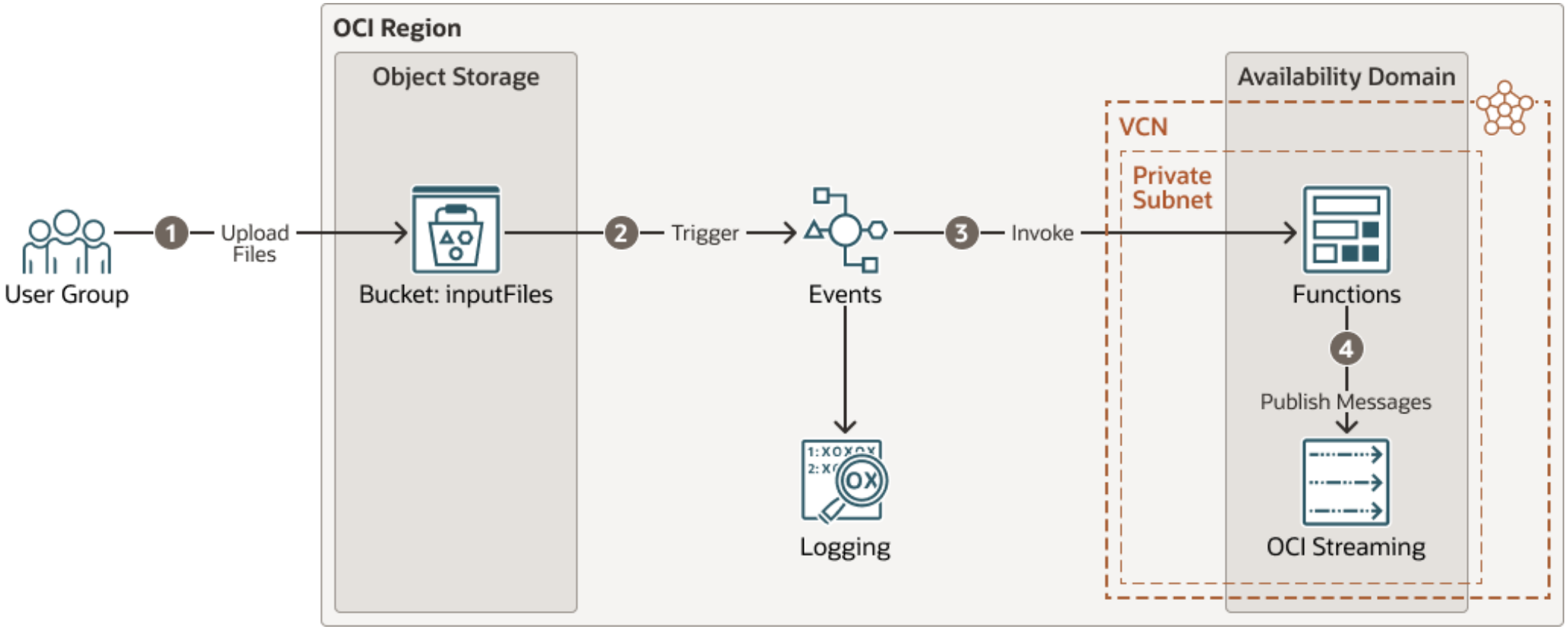
Key Benefits

- **Event-Driven**: Automatically trigger workflows.
- **Cost-Effective**: Pay only for what you use.
- **Scalable**: Automatically adjusts to traffic.
- **Secure**: Integrated with OCI's Identity and Access Management (IAM).
- **Language Flexibility**: Use your favorite programming language.
- **Integration**: Works seamlessly with OCI services like API Gateway, Object Storage, Events, and more.

Why Are OCI Functions Fun?

- **Freedom from Infrastructure**: Focus on code, not servers.
- **Easy to Experiment**: Build and test prototypes quickly.
- **Event-Driven Magic**: Automate workflows by triggering functions with events.
- **Cost Efficiency**: Only pay for what you use – perfect for small, experimental projects or large-scale production workloads.

OCI Function Example



OCI Provided Functions

Cloud | Search resources, services, documentation, and Marketplace | US East (Ashburn)

Functions

Applications

Pre-Built Functions

Filters

Triggers

Any

Tag filters add | clear

no tag filters applied

Pre-Built Functions

Discover ready-to-use Functions that are pre-built to execute certain tasks or actions across OCI services and deploy them with one click without writing any code.

Search...

OpenSearch RAG Pipeline Creator

This pre-built function executes commands within an OpenSearch cluster to create a RAG pipeline to be used for conversational search. The function will add necessary settings to the cluster, create a model group, connector, and pipeline, and register and deploy a…

APM Log Sender

This Pre-built Function (PBF) can be used in Oracle Cloud Infrastructure (OCI) Connector Hub to forward service logs from the OCI Logging service to OCI Application Performance Monitoring (APM). You can create a function from this PBF and configure it as a…

Cost Reports FOCUS converter

This pre-built function converts the OCI generated cost report csvs to the FinOps Open Cost and Usage Specification (FOCUS) schema. Your proprietary cost reports will be converted to FOCUS format and stored in an Object Store destination bucket that you specif…

Database Secret Rotation with Wallet

This Pre-Built Function (PBF) streamlines the rotation process by generating and updating credentials in Secrets in Vault (SiV), enhancing security with regular changes to access credentials. It employs MTLS connection, and can rotate both admin and non-adm…

Database Secret Rotation without Wallet

This Pre-Built Function (PBF) streamlines the rotation process by generating and updating credentials in the Secrets in Vault (SiV), enhancing security through regular changes to access credentials. Using TLS, it establishes connections to rotate both admin and no…

Document Generator

This Pre-Built Function (PBF) generates documents in an Oracle Cloud Infrastructure (OCI) Object Storage bucket based on provided JSON data and an Office template document stored in Object Storage.

Object Storage File Extractor

This Pre-built Function (PBF) reads a zip file from an Oracle Cloud Infrastructure (OCI) Object Storage bucket and extracts it to the specified target bucket.

Object Storage File Zip

This Pre-Built Function (PBF) is used to zip files stored in an Oracle Cloud Infrastructure (OCI) Object storage bucket and save the zip file to the specified target bucket

Media Workflow Job Spawner

This Pre-Built function (PBF) triggers a workflow job for the given Media workflow when video content is uploaded to Object storage bucket. To set up this automation - 1) Deploy this PBF as a Function in your

Ideas For OCI Function

1. Automated File Processing

- Trigger functions when files are uploaded to Object Storage (e.g., compress or process images).

2. Real-Time Data Transformation

- Process and transform streaming data for analytics or monitoring pipelines.

3. Chatbot Backend

- Run serverless logic to power chatbots or customer support workflows.

4. Notifications and Alerts

- Send notifications via email, SMS, or Slack when specific events occur.

5. Scheduled Tasks

- Automate periodic tasks like database cleanups or data backups.

6. API Integration

- Build lightweight APIs to connect and integrate with third-party services.

7. IoT Data Processing

- Process sensor data from Internet of Things (IoT) devices quickly and scalably.

8. Machine Learning Inference

- Deploy models for real-time predictions without managing servers.

9. Serverless Webhooks

- Respond to webhooks for CI/CD pipelines or application events.

10. Fraud Detection

- Analyze transactions in real-time to identify fraudulent activities.

11. Content Moderation

- Use AI models or rules to filter and moderate uploaded content.

12. DevOps Automation

- Automate deployments, health checks, or scaling workflows in DevOps pipelines.

GitHub Examples And Demo Code

Description	Python	Java
Hello World	sample	sample
List OCI Compute instances	sample	sample
Control OCI Compute instances (start/stop/status)	sample	
List OCI compartments	sample	
List objects in OCI Object Storage	sample	sample
Read an object in OCI Object Storage	sample	sample
Create an object in OCI Object Storage	sample	sample
Create a PAR in OCI Object Storage	sample	
Copy object from one OCI Object Storage bucket to another	sample	
Display an OCI Cloud Event	sample	
Invoke another Function using the OCI SDK	sample	
Run a SQL statement against Autonomous DB using ORDS	sample	
Run a SQL statement against Autonomous DB using DB Client	sample	
Publish a notification using ONS	sample	
Send an email using Email Delivery Service	sample	
Decrypt cipher using Vault keys	sample	
Get a secret from Vault	sample	
Write IAM policies that enables Functions in a te Screenshot ess resources in other		sample

https://github.com/oracle-samples/oracle-functions-samples

Mike Miller's OCI Functions

- Drain IAM dormant users
- IAM Status for SSO
- Create iCinga Alerts and ServiceNow incidents using specific CMDB CI info
- Backup up OIC instances on a schedule
- Backup of OAC instances on a schedule
- CVE Reporting for EBS (Full Stack) using MySQL DB and Object Storage
- OCI Inventory and delta reporting
- OCI Vault backups
- Sweep with criteria for CloudGuard and Cloud Advisor raise alarms and tickets

Are Functions Free? No. But There Is a Free Tier

Free Tier

- Execution Time of 400,000 GB-seconds per month
- 2 million requests per month

Paid Tier

- Execution Time at $0.00001417 per gigabyte-second for execution time.
- Requests are charged at $0.20 per million.

Example

•Requests: 500,000 per month.
•Execution Time: 2 seconds per request.
•Allocated Memory: 512 MB (0.5 GB).

Step 1: Execution Cost Memory (GB)} Execution Time 2 seconds x 0.5 GB x 0.00001667 x 500,000 = $8.34

Step 2: Request Cost 500,000 x 0.0000002 = $0.10

Total Monthly Cost = $8.34 + $0.10 = $8.44

OCI Function Cost Factors

1. **Memory Allocation**:
    1. Higher memory allocation increases execution cost.
    2. Choose the minimum memory your function needs to run efficiently.
2. **Execution Time**:
    1. Optimize your code to reduce execution time.
3. **Number of Requests**:
    1. Reduce unnecessary function calls by optimizing triggers.

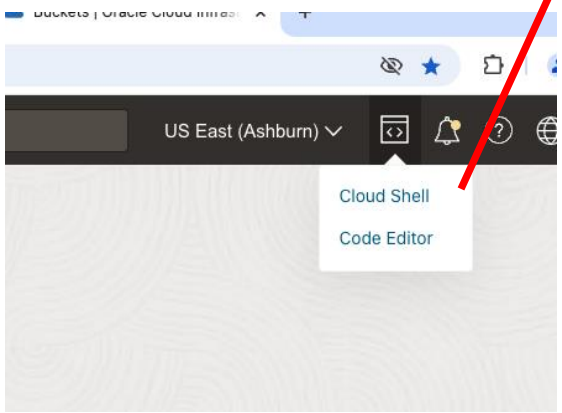
Creating And Deploying Functions

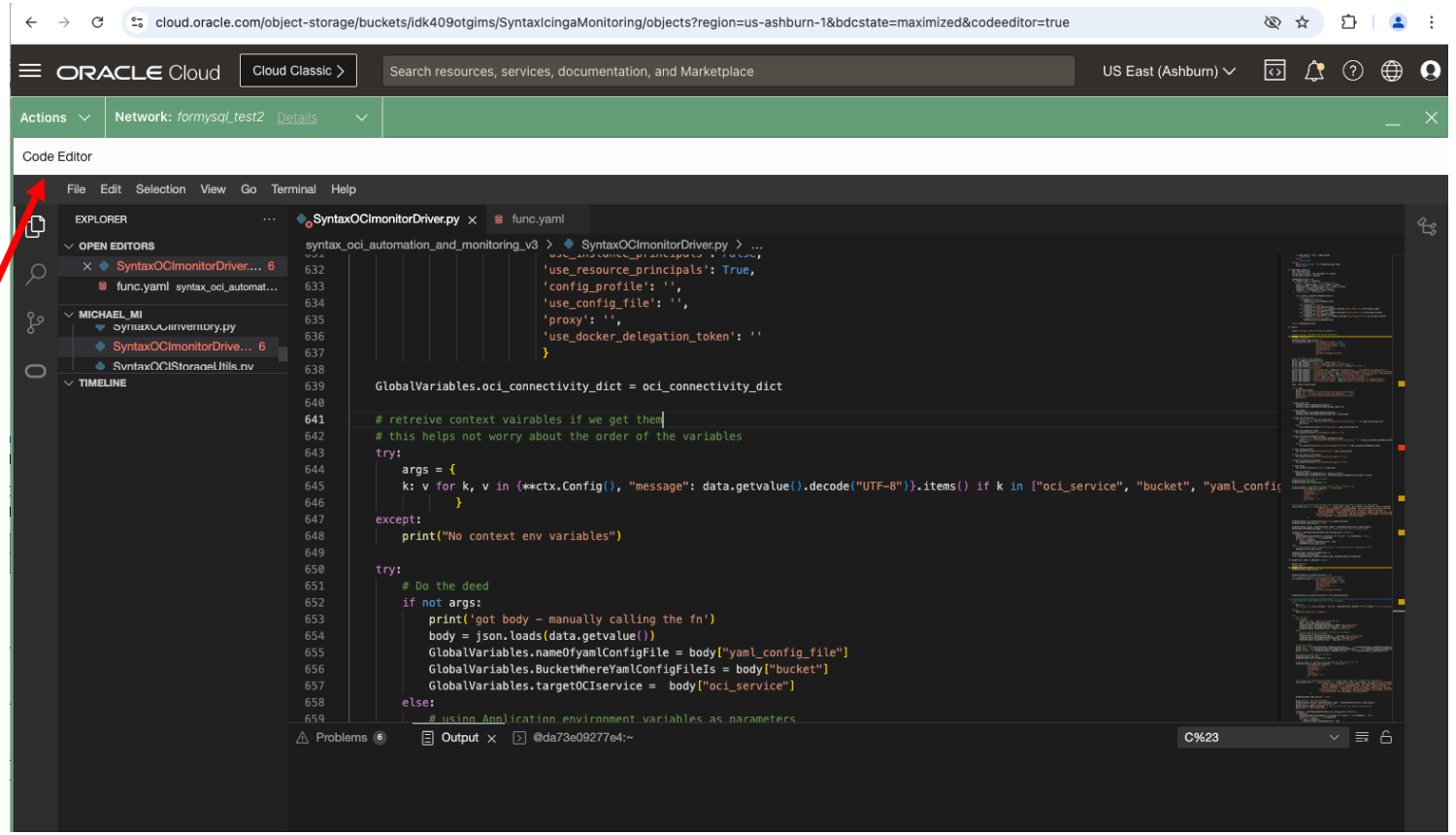
Following The Instructions Is Easier Than Not

Step One: Use the OCI Cloud Editor To Write Your Code

Use Cloud Shell/Editor

- Delegation tokens are simpler than working through authentication and authorization issues off your laptop
- Test your code successfully from Cloud Shell/Editor as your first step





This Is The Cloud Shell

Cloud | Search resources, services, documentation, and Marketplace | US East (Ashburn)

Actions | Network: *formysql_test2* Details

Cloud Shell

```
Welcome to Oracle Cloud Shell.

Cloud Shell now runs on Oracle Linux 8. For more information, see the release notes. For help, contact Oracle Support.

Your Cloud Shell machine comes with 5GB of storage for your home directory. Your Cloud Shell (machine and home directory) are located in: US East (Ashburn).
You are using Cloud Shell in tenancy syntaxoci as OCI local user Michael.Miller@syntax.com

Type `help` for more info.
Michael_Mi@cloudshell:~ (us-ashburn-1)$ ls
a_new_venv  syntax_oci_automation_and_monitoring_v10  syntax_oic_automation
Michael_Mi@cloudshell:~ (us-ashburn-1)$
```

This Is The Cloud Code Editor

Cloud | Search resources, services, documentation, and Marketplace | US East (Ashburn)

Actions | Network: *formysql_test2* Details

Code Editor

File Edit Selection View Go Terminal Help

EXPLORER

OPEN EDITORS

SYNTAX_OIC_AUTOMATION

- __pycache__
- Dockerfile3828427475
- func.yaml
- requirements.txt
- SyntaxGlobalVariablesAnd...
- SyntaxIcingaUtils.py
- SyntaxIntegrationCloudB2...
- SyntaxIntegrationCloudFTP...
- SyntaxIntegrationCloudFTP...
- SyntaxIntegrationCloudInte...
- SyntaxIntegrationCloudInte...
- SyntaxOCIconnectivity.py
- SyntaxOCImonitorDriver.py
- SyntaxOCIvaultUtils.py

SyntaxOCImonitorDriver.py | SyntaxIntegrationCloudIntegrationError.py

SyntaxIntegrationCloudIntegrationError.py

```
234                         'integration_cloud']),
235                     'description': ' OIC Integration Errors Detected by OCI Monitoring - Log into production to investigate for cli
236                     'title': 'OIC Integration Errors Detected by OCI Monitoring',
237                     'severity': 'High'
238                     }
239         SyntaxIcingaUtils.raise_icinga_alert(the_alert)
240
241     except oci.exceptions.ServiceError as e:
242         the_alert = {'configuration_item': SyntaxGlobalVariablesAndUtils.GetConfigFromYAML(['icinga','configuration_items',
243                         'integration_cloud']),
244                     'description': ' Error Pushing OIC Errors '+str(e.message),
245                     'title': 'Error Pushing OIC Errors',
246                     'severity': 'High'
247                     }
248         SyntaxIcingaUtils.raise_icinga_alert(the_alert)
```

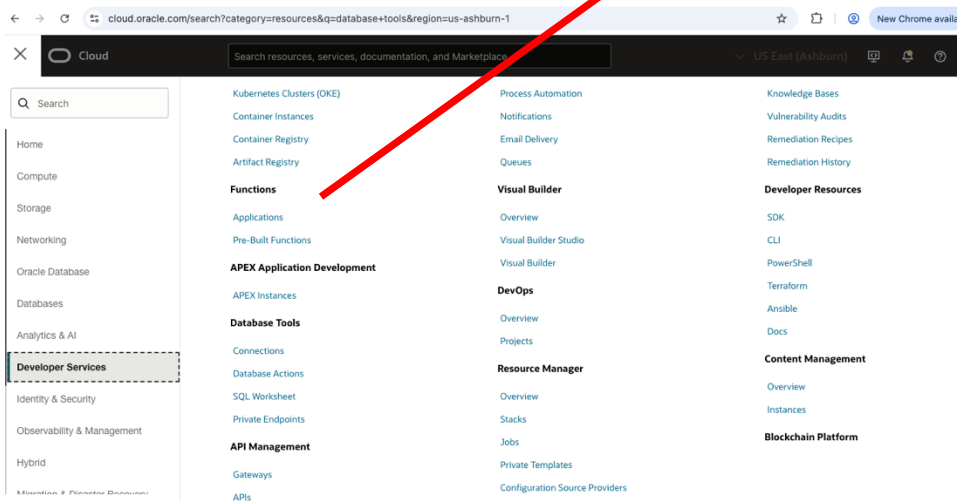
Problems | @49fa11f7f9ad:~/syntax_oic_automation

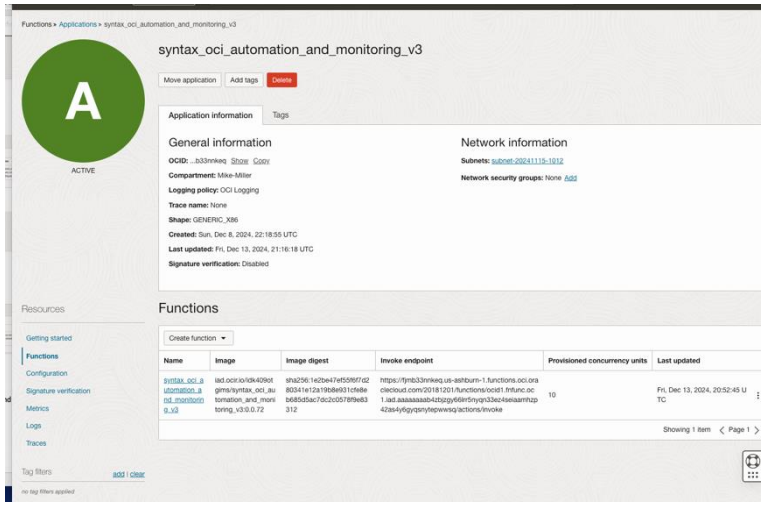
```
Michael_Mi@codeeditor:syntax_oic_automation (us-ashburn-1)$
```

OATUG OHUG ASCEND 2025

Step Two: Following The Instructions Is MUCH Easier Than Not

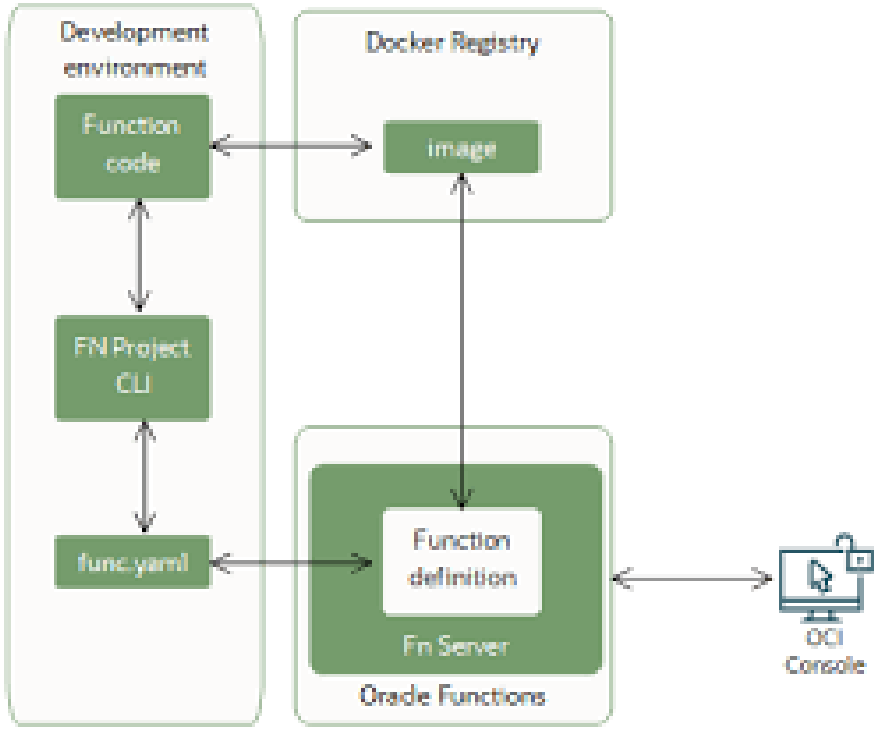
1. Write and test your code
2. Developer Tools --> Application --> Function





Deploying Functions: Mike's Cheat Sheet

1. Open OCI Cloud Shell
2. Source and activate the venv
3. fn init --runtime python syntax_oci_function_demo
4. Change the directory into the newly created directory with the same name as the function
5. Copy your Python code into this directory
6. Edit the func.yaml file.
7. pip freeze > requirements.txt
8. Remove setup.py and any working or junk files
9. fn list context
10. fn use context us-ashburn-1
11. fn update context oracle.compartment-id ocid1.compartment.oc1..aaaaaaaai3wyta3bzc467dw4hjhjvqiesqqs5oewhxokbbdnstbaug5tqoja
12. fn update context registry iad.ocir.io/idk409otgims
13. docker login -u 'idk409otgims/Michael.Miller@syntax.com' iad.ocir.io
14. Type my password
15. fn -v deploy –app syntax_oci_function_demo



When Things Go Wrong Deploying a Function

- Cloud Shell editor is better than a laptop
- Manually run your function from Cloud Editor or Shell
- Confirm using the correct venv
- Review func.yaml file
  - Check your spelling carefully
  - Ensure space **AFTER** your *.py script and the function ‘handler’

    entrypoint: /python/bin/fdk /function/SyntaxOCImonitorDriver.py handler

- Requirements.txt must be correct
  - Use Freeze to recreate if needed: pip freeze > requirements.txt
- Don’t copy in the setup.py file from your laptop into the deployment directory
- Confirm version of Python: 3.11 for Functions and 3.8 for Cloud Shell/Editor vs your laptop
- Get no space left on device. The cloud shell is limited to 5 GB! If all else fails: csreset -a

Don’t Mess Up the Func.yaml File

Fn init will generate a a directory with a demo file func.yaml that you need to edit

- Get the memory correct
- Match the name exactly
- Note the spaces between the python binary, your script and the handler entry function

fn init --runtime python <name of your function>

func.yaml

```
schema_version: 20180708
name: syntax_oic_automation
version: 0.0.1
runtime: python
build_image: fnproject/python:3.11-dev
run_image: fnproject/python:3.11
entrypoint: /python/bin/fdk /function/SyntaxOCImonitorDriver.py handler
memory: 3072
```

The Handler Function

If you want to use OCI application parameters – *read this slide carefully* – you need an entry point function with two parameters:

- ctx
- data

SyntaxOCImonitorDriver.py scratch_6.py

```python
def handler(ctx, data: io.BytesIO = None):
    try:
        args = {
        k: v for k, v in {**ctx.Config(),
                          "message": data.getvalue().decode("UTF-8")}.items()
                               if k in ["oci_service", "bucket", "yaml_config_file"]
                }
    except:
        print("No context env variables")
    try:
        GlobalVariables.nameOfyamlConfigFile = args.get("yaml_config_file")
        GlobalVariables.BucketWhereYamlConfigFileIs = args.get("bucket")
        GlobalVariables.targetOCIservice = args.get("oci_service")

        if GlobalVariables.targetOCIservice:
            DoTheWork(valid_inputs_dict)
        else:
            print('Nothingn to do')
        result = str('Success')
        result_text = "Function result: "+result
    except (Exception, ValueError) as ex:
        result = str('Fell into error exception in main handler')
        logging.getLogger().info('Function error in handler: ' + str(ex))

    # logging.getLogger().info("Inside Python Hello World function")
    print('Done handling the handler')
    return response.Response(
        ctx, response_data=json.dumps(
            {"message": result_text}),
        headers={"Content-Type": "application/json"}
    )
```

OCI Function Limits

OCI Functions have regional limits

Most important Functions will execute for a maximum of 500 seconds (five minutes)!

source	Limit Name	Description	Scope	Oracle Universal Credits	Pay As You Go or Trial
Applications	application-count	Number of applications	Region	20	10
Functions	function-count	Number of functions	Region	500	50
Total memory for concurrent function execution	total-concurrency-mb	Overall total amount of memory available to allocate for concurrent execution of all functions in the region. Actual memory usage is always less than or equal to this amount.	Region	60 GB(180 GB for a region with three availability domains)	60 GB(180 GB for a region with three availability domains)
Total memory for provisioned concurrency	provisioned-concurrency-mb	Overall total amount of memory available to allocate for provisioned concurrency for all functions in the region. Actual memory usage is always less than or equal to this amount.	Region	40 GB	40 GB

https://docs.oracle.com/en-us/iaas/Content/General/Concepts/servicelimits.htm#Functions_Limits

Functions And Temp Files

Deployed OCI Functions can access the file system of the container in which it's running as follows:

- the function can read files from all directories
- the function can write files to the /tmp directory

Maximum memory threshold for the function (MB)	Maximum allowed size of /tmp (MB)	Maximum allowed number of files (inodes) in /tmp
128 MB	32 MB	1,024
256 MB	64 MB	2,048
512 MB	128 MB	4,096
1024 MB	256 MB	8,192
2048 MB	512 MB	16,384
3072 MB	768 MB	24,576

https://docs.oracle.com/en-us/iaas/Content/Functions/Tasks/functionsaccessinglocalfilesystem.htm

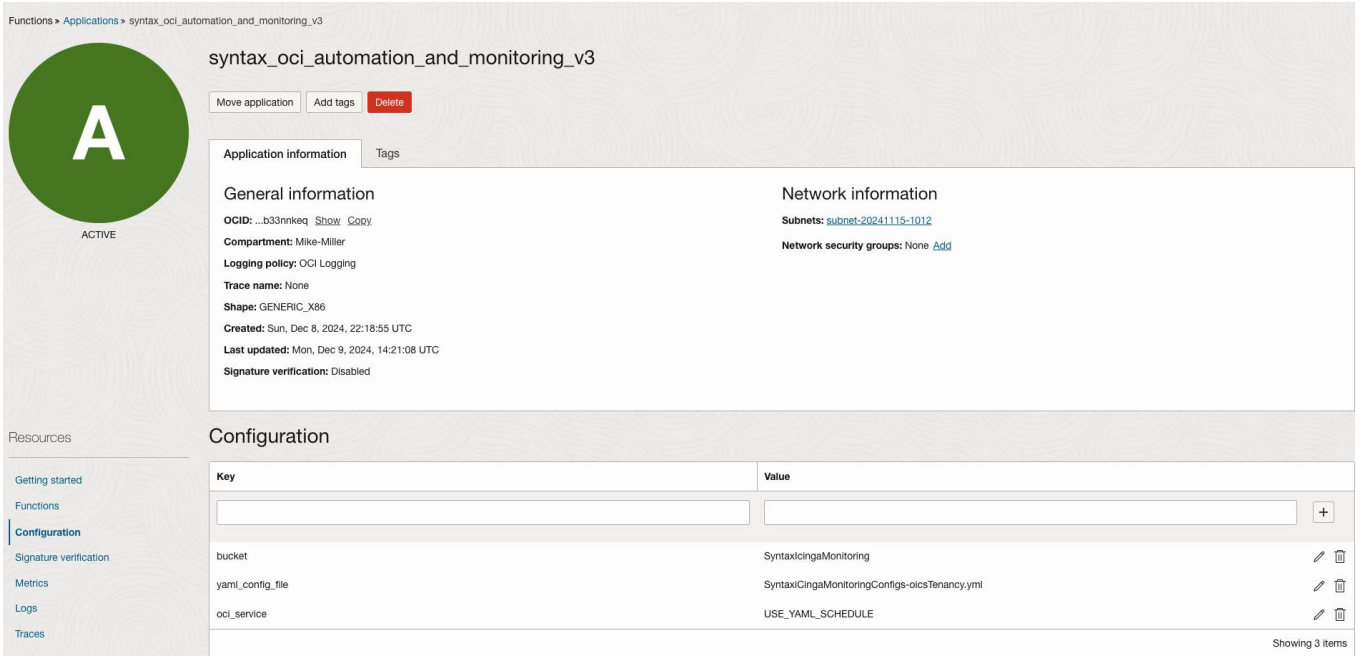
Define Function Parameters

Define (static0 parameters to make it easier to invoke your function:

fn invoke syntax_oic_automation syntax_oic_automation

Save this for syntax calling with dynamic not static parameters:

```
echo -n
'{"yaml_config_file":"SyntaxiCingaMonitorin
gConfigs-oicsTenancy.yml"
,"bucket":"SyntaxIcingaMonitoring",
"oci_service": "IAM"}'| fn invoke
syntax_oci_automation_and_monitoring_v3
syntax_oci_automation_and_monitoring_v3
```



Example: Notification Topic Calling A Function

Cloud | Search resources, services, documentation, and Marketplace | US East (Ashburn)

Notifications » Topics » Topic Details

T

ACTIVE

testIcingaWithFunction

Publish Message | Move Resource | Add tags | Delete

Topic Information | Tags

Description:

OCID: ...xqgibl37hq Show Copy

Short Topic ID: - Show Copy

Compartment: syntaxoci (root)/Architects/Mike-Miller

Created: Tue, Apr 22, 2025, 23:17:08 UTC

Resources

- **Subscriptions**
- Metrics

List scope

Compartment

Subscriptions

Create Subscription

Subscription OCID	State	Protocol	Endpoint	Created
ocid.....qnpq7qfq	Active	Function	**Function:** syntax_oic_automation **Application:** syntax_oic_automation	Tue, Apr 22, 2025, 23:17:34 UTC

Example: OCI Resource Scheduler Calling A Function

Cloud | Search resources, services, documentation, and Marketplace | US East (Ashburn)

Resource scheduler » MM Hourly

S

ACTIVE

MM Hourly

Test of Hourly

Edit schedule | Clone schedule | Disable schedule | Add tags | Delete schedule

Schedule details | Tags

OCID: ...xxjdzdmq Show Copy

Compartment: syntaxoci (root)

Recurrence details: Every hour at 13:35 UTC

Action: Start

Resources filtered by: 1 resource

Date created: Mon, 21 Apr 2025 20:09:34 UTC

Start date: Tue, 22 Apr 2025 13:35:00 UTC

End date: Wed, 30 Apr 2025 00:00:00 UTC

Last run date: Fri, 25 Apr 2025 15:35:17 UTC

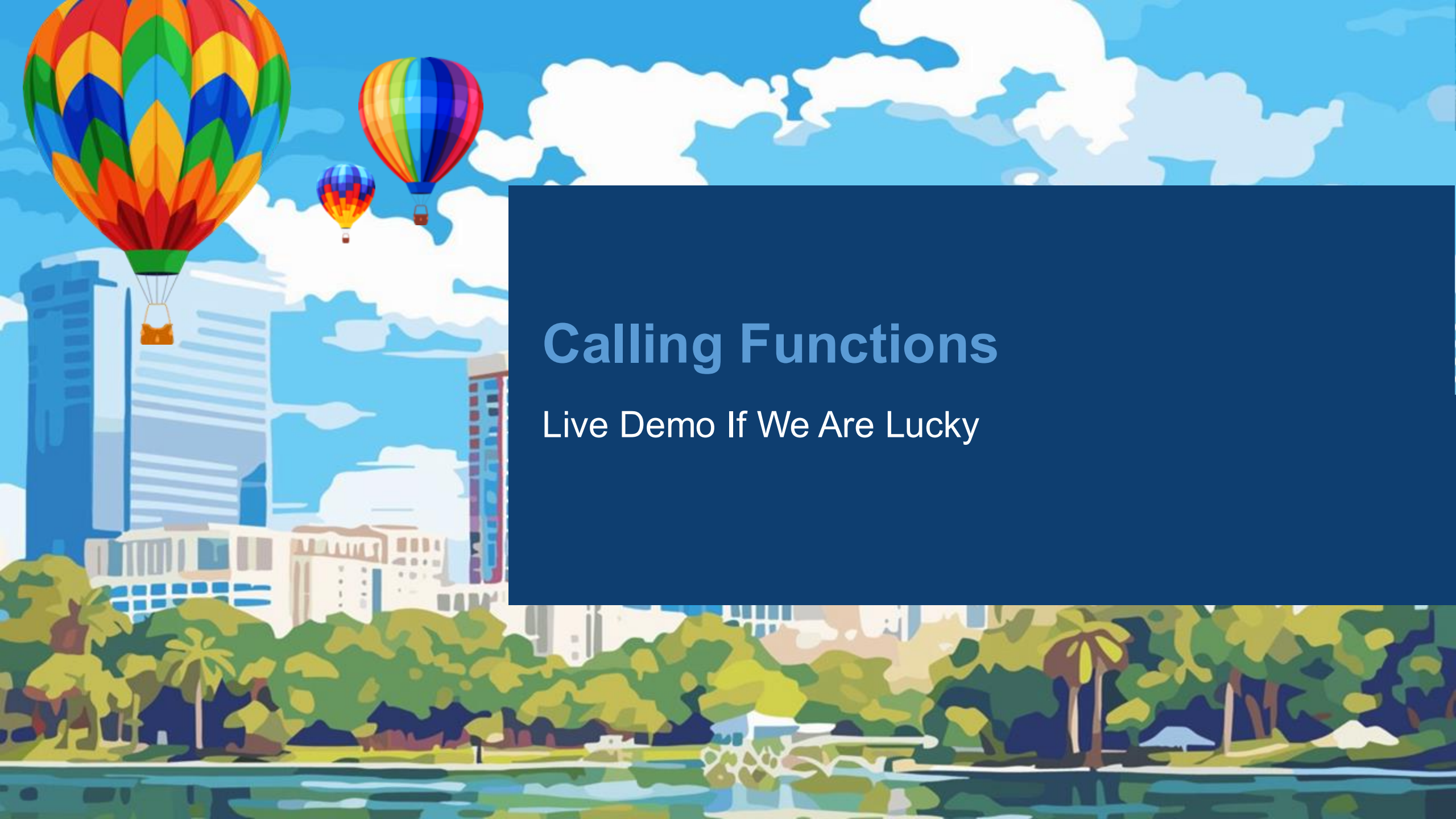
Next run date: Fri, 25 Apr 2025 16:35:00 UTC

Resources

Work requests

Work requests

Run date	Operation type	Resources	Status	Duration
Fri, Apr 25, 2025, 15:35:17 UTC	Start resource	1 resource	Succeeded	1 second
Fri, Apr 25, 2025, 14:35:12 UTC	Start resource	1 resource	Succeeded	801 ms
Fri, Apr 25, 2025, 13:35:15 UTC	Start resource	1 resource	Succeeded	703 ms

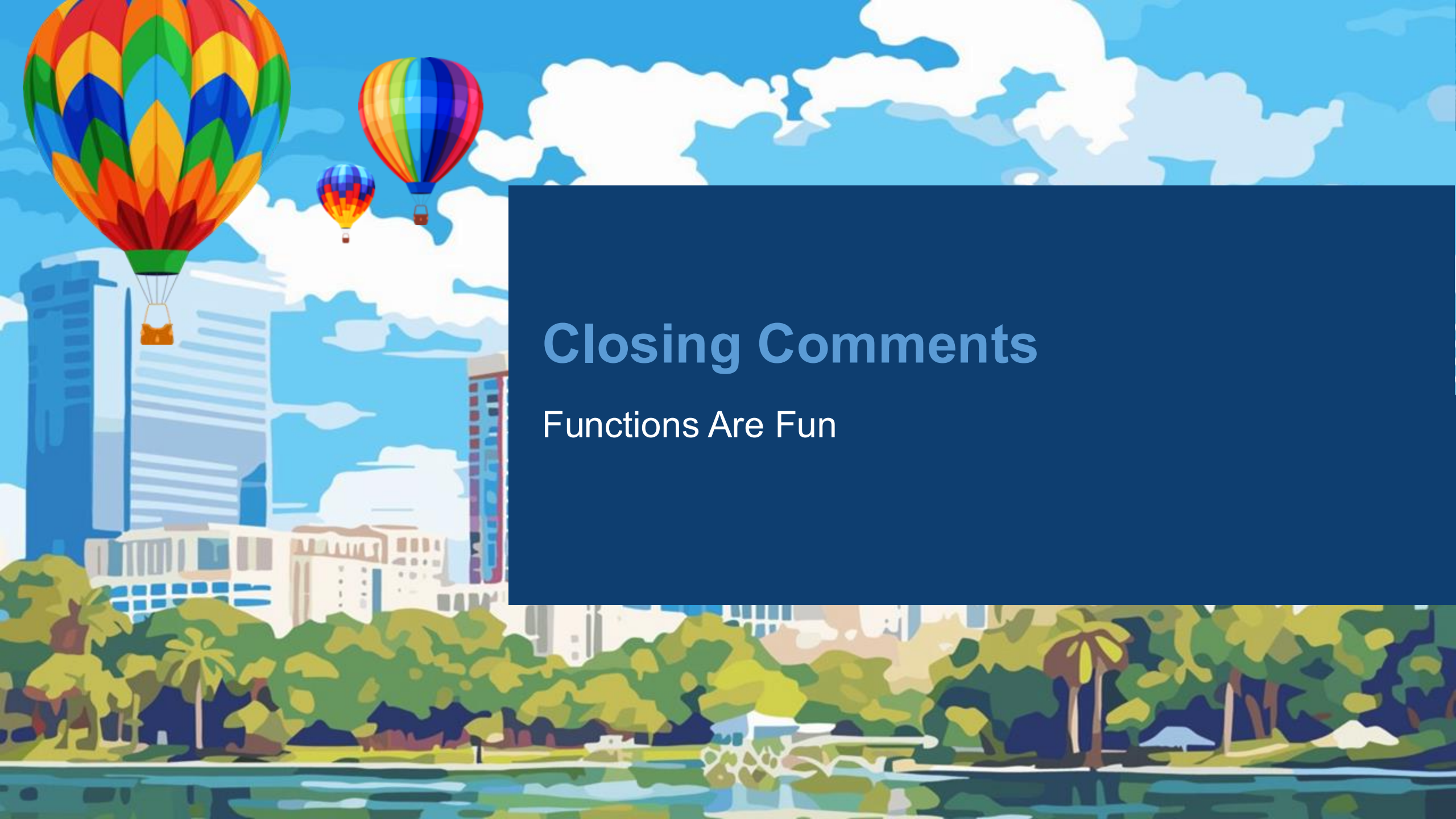


Calling Functions

Live Demo If We Are Lucky

Live Demo From

- Deploy a function
- Then call it



Closing Comments

Functions Are Fun

OCI Functions Are Fun

Have script? Make it a function!

Remember

- Use the Cloud Code Editor to write code
- Test first: 5 minutes and done!
- Make sure requirements.txt is correct
- Keep in mind the OCI Resource scheduler

Why are OCI Functions fun?

- **Zero Server Hassle**: Focus on writing code, not managing infrastructure.
- **Scalable**: Automatically adjusts to traffic.
- **Seamless Integration**: Easily connect with other OCI services.
  - **Event-Driven**: Automatically trigger workflows.
  - **Cost-Effective**: Pay only for what you use.
  - **Secure**: Integrated with OCI's Identity and Access Management (IAM).

Thank You For Attending!

Please complete the session survey in the conference app.

ASCEND 2025 | OATUG ORACLE APPLICATIONS & TECHNOLOGY USERS GROUP | OHUG ORACLE HCM USERS GROUP

Q&A

Michael.Miller@syntax.com

ASCEND 2025 | OATUG ORACLE APPLICATIONS & TECHNOLOGY USERS GROUP | OHUG ORACLE HCM USERS GROUP