



aws SUMMIT

LONDON | JUNE 7, 2023

DEV201

Maximizing Developer Productivity with Amazon CodeWhisperer

Om Prakash Jha (he/him)
Senior Solutions Architect
AWS

Laura Al-Richane (she/her)
Solutions Architect
AWS

Pandurang Kamat (he/him)
Chief Technology Officer
Persistent Systems



Sneak peek into an engineer's life



Engineer



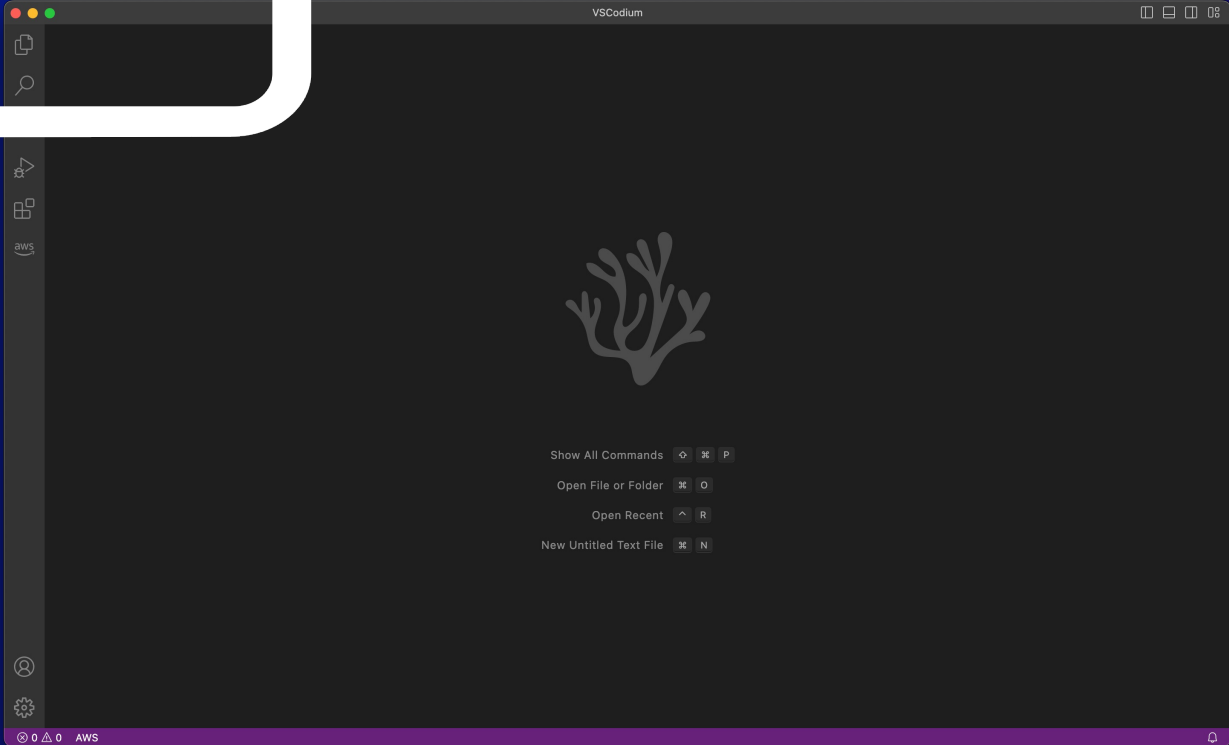
Product Manager

Sneak peek into an engineer's life



Engineer

Sure, no problem!

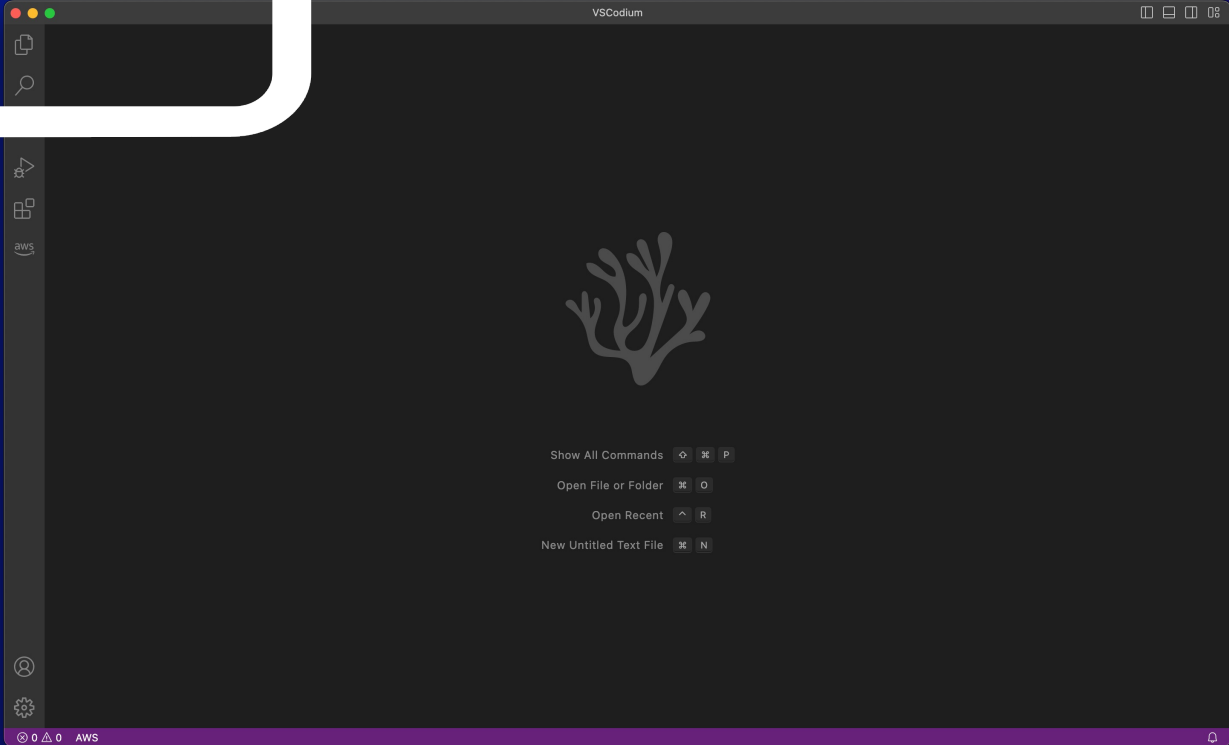


Sneak peek into an engineer's life



Engineer

Hmm, but where do I start?



Sneak peek into an engineer's life

The screenshot shows the AWS Developer Guide for Amazon Rekognition. The main heading is "What is Amazon Rekognition?". The page is divided into sections: "On this page", "Amazon Rekognition and HIPAA eligibility", "Are you a first-time Amazon Rekognition user?", "Common use cases for using Amazon Rekognition include the following:", "Searchable image and video libraries", "Face-based user verification", "Detection of Personal Protective Equipment", "Sentiment and demographic analysis", and "Facial Search".

The screenshot shows the AWS Boto3 1.26.96 documentation page. It features the AWS logo at the top, a search bar, and a "Feedback" section with the text: "Do you have a suggestion to improve this website or boto3? Give us feedback." Below the feedback section are links for "Quickstart" and "A Sample Tutorial".

The screenshot shows the AWS Rekognition Client documentation page. The main heading is "Rekognition Client". Below the heading is the text: "A low-level client representing Amazon Rekognition". It also includes a list of "ON THIS PAGE" items: "Client", "Rekognition.Client", "Paginators", and "Waiters".

Amazon Rekognition Image

- CompareFaces
- CreateCollection
- DeleteCollection
- DeleteFaces
- DescribeCollection
- DetectFaces
- DetectLabels
- DetectModerationLab
- DetectProtectiveEquip
- DetectText
- GetCelebrityInfo
- IndexFaces
- ListCollections
- ListFaces
- RecognizeCelebrities
- SearchFaces
- SearchFacesByImage

Amazon Rekognition

- CopyProjectVersion
- CreateDataset
- CreateProject
- CreateProjectVersion

The screenshot shows a forum thread on AWS re:Post titled "Custom train videos in Rekognition". The thread is started by user "AWS-User-3217907" and includes a question about content moderation. It has 3 answers and relevant content links.

The screenshot shows a GitHub repository for "aws-samples/amazon-rekognition-code-samples". It displays a Jupyter Notebook titled "Celebrity Recognition using Amazon Rekognition". The notebook code includes sections for "Initialize stuff", "Current AWS Region", "Fair clients", "S3 bucket", and "Create temporary directory".

```
# Initialize Notebook
import boto3
from IPython.display import HTML, display, Image as Image
from PIL import Image, ImageDraw, ImageFilter
import time
import os

# Current AWS Region. Use this to choose corresponding S3 bucket with sample content
mySession = boto3.session.Session()
awsRegion = mySession.region_name

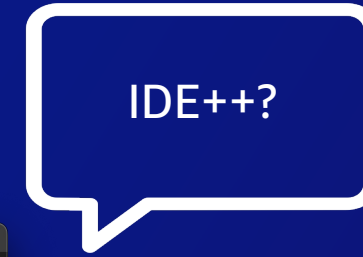
# Fair clients
rekognition = boto3.client('rekognition')
s3 = boto3.client('s3')

# S3 bucket that contains sample images and videos
# We are providing sample images and videos in this bucket so
# you do not have to manually download/upload test images and videos.
bucketName = "aws-rek-lmsersinday" + awsRegion

# Create temporary directory
# This directory is not needed to call Rekognition APIs.
# We will only use this directory to download images from S3 bucket and draw bounding boxes
# around recognized celebrities to show them here in the notebook.
mkdir mkdir
tempFolder = "step1"
```



Wouldn't it be great if...



```
VSCodeium
# Write a function to upload a file to S3.
def upload_file_to_s3(file_name, bucket_name, object_name):
    """
    Uploads a file to an S3 bucket

    :param file_name: File to upload
    :param bucket_name: Bucket to upload to
    :param object_name: S3 object name. If none then file_name is used
    :return: True if file was uploaded, else False
    """

    # Upload the file
    s3_client = boto3.client('s3',
                             aws_access_key_id=AWS_ACCESS_KEY_ID,
                             aws_secret_access_key=AWS_SECRET_ACCESS_KEY,
                             region_name=AWS_REGION_NAME)

    try:
        s3_client.upload_file(file_name, bucket_name, object_name)
        print(f'File {file_name} uploaded to S3 bucket {bucket_name} as {object_name}')
        return True
    except FileNotFoundError:
        print(f'File {file_name} not found')
```

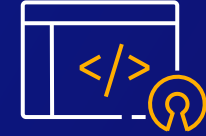
Quickly building reliable code is challenging



Multiple programming languages



Time spent on undifferentiated code



Appropriate use of open source



Losing focus through context switches



Time spent learning technologies, APIs, and best practices



Application security

Introducing Amazon CodeWhisperer

BUILD APPLICATIONS FASTER AND MORE SECURELY WITH YOUR AI CODING COMPANION



- AI coding companion integrated in your IDE
- Provides code recommendations based on comments in natural language and prior code

CodeWhisperer: AI Coding Companion

Provides code recommendations based on natural language comments and contextual information such as prior code

```
# Write a function to upload a file to S3.
def upload_file_to_s3(file_name, bucket_name, object_name):
    """
    Uploads a file to an S3 bucket

    :param file_name: File to upload
    :param bucket_name: Bucket to upload to
    :param object_name: S3 object name. If none then file_name is used
    :return: True if file was uploaded, else False
    """

    # Upload the file
    s3_client = boto3.client('s3',
                             aws_access_key_id=AWS_ACCESS_KEY_ID,
                             aws_secret_access_key=AWS_SECRET_ACCESS_KEY,
                             region_name=AWS_REGION_NAME)

    try:
        s3_client.upload_file(file_name, bucket_name, object_name)
        print(f'File {file_name} uploaded to S3 bucket {bucket_name} as {object_name}')
        return True
    except FileNotFoundError:
        print(f'File {file_name} not found')
```

CodeWhisperer: AI Coding Companion

#Upload a file to S3

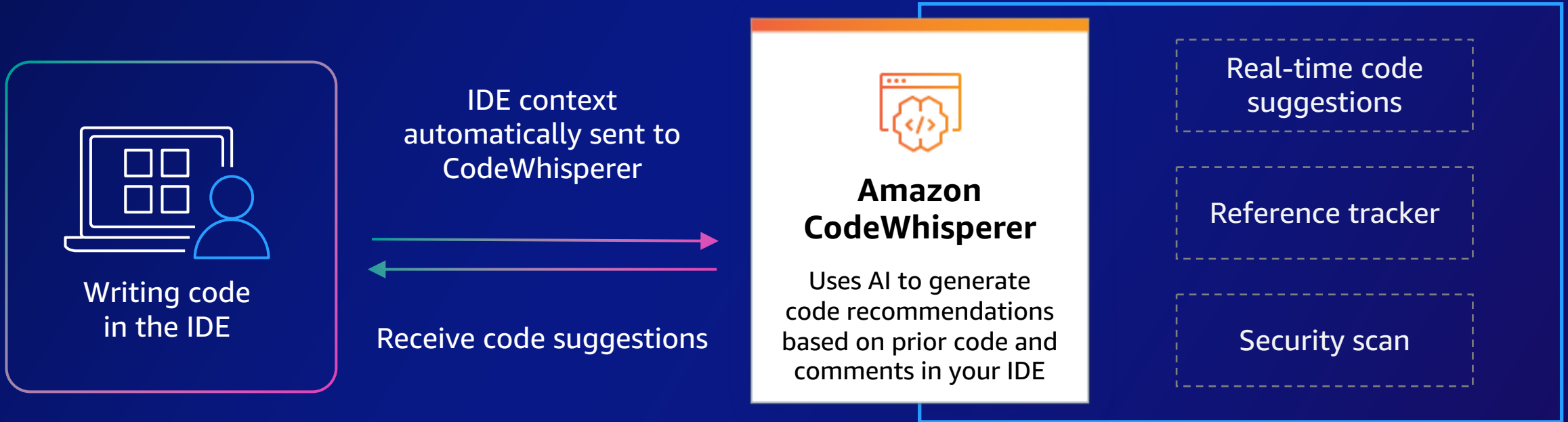
```
# Write a function to upload a file to S3.
def upload_file_to_s3(file_name, bucket_name, object_name):
    """
    Uploads a file to an S3 bucket

    :param file_name: File to upload
    :param bucket_name: Bucket to upload to
    :param object_name: S3 object name. If none then file_name is used
    :return: True if file was uploaded, else False
    """

    # Upload the file
    s3_client = boto3.client('s3',
                             aws_access_key_id=AWS_ACCESS_KEY_ID,
                             aws_secret_access_key=AWS_SECRET_ACCESS_KEY,
                             region_name=AWS_REGION_NAME)

    try:
        s3_client.upload_file(file_name, bucket_name, object_name)
        print(f'File {file_name} uploaded to S3 bucket {bucket_name} as {object_name}')
        return True
    except FileNotFoundError:
        print(f'File {file_name} not found')
```

CodeWhisperer: How it works

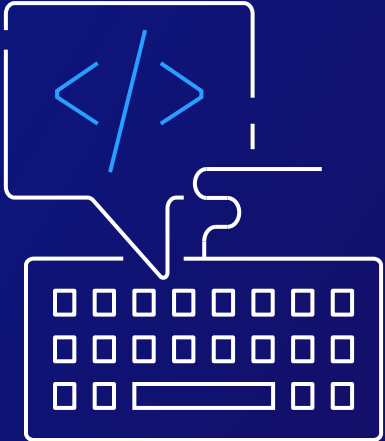


CodeWhisperer features

Supported ecosystem
programming languages



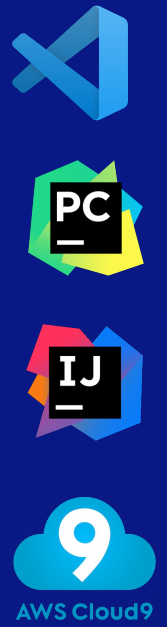
Go, Rust, PHP,
Ruby, Kotlin,
C, C++,
Shell scripting,
SQL, and Scala



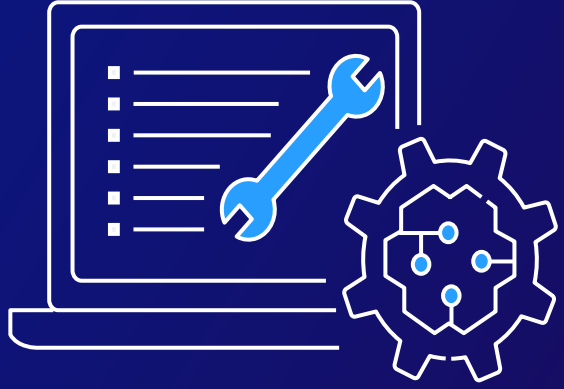
CodeWhisperer features

Supported ecosystem

IDEs

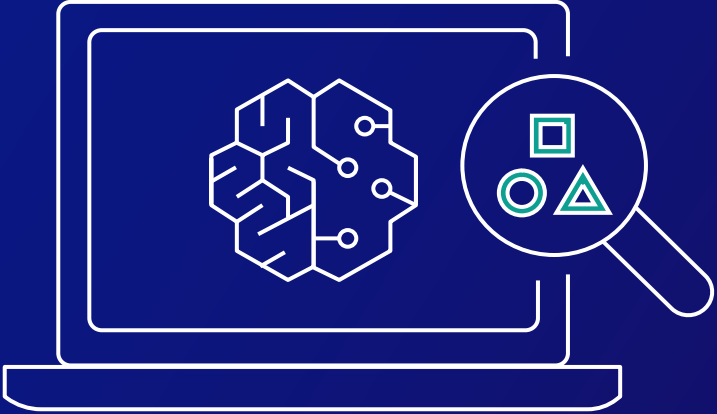


CLion, GoLand,
WebStorm,
Rider, PhpStorm,
RubyMine, and
DataGrip



CodeWhisperer features

Reference tracker



CodeWhisperer features

Security scan



Demo





algorithm.py

1

I





email-validation.py

1 |



unit-test.py

1

I



recognize.py

1



We are Persistent.

We are a trusted **Digital Engineering** and **Enterprise Modernization** partner.

33+ years of leadership in software engineering and digital transformation.

Highest customer experience scores of any company in IT industry, as rated by ISG.

Boutique mindset focused on enterprise clients moving their digital presence to the cloud.

Diverse, open and innovative business partner ecosystem for maximum adaptability.

1990
Founded

\$1,036M
FY23 Annual Revenue

350+
Clients Annually

22,750+
Employees

21
Countries



Persistent Systems: Amazon CodeWhisperer Experience

What our developers love about Amazon CodeWhisperer

**Improved
Productivity**

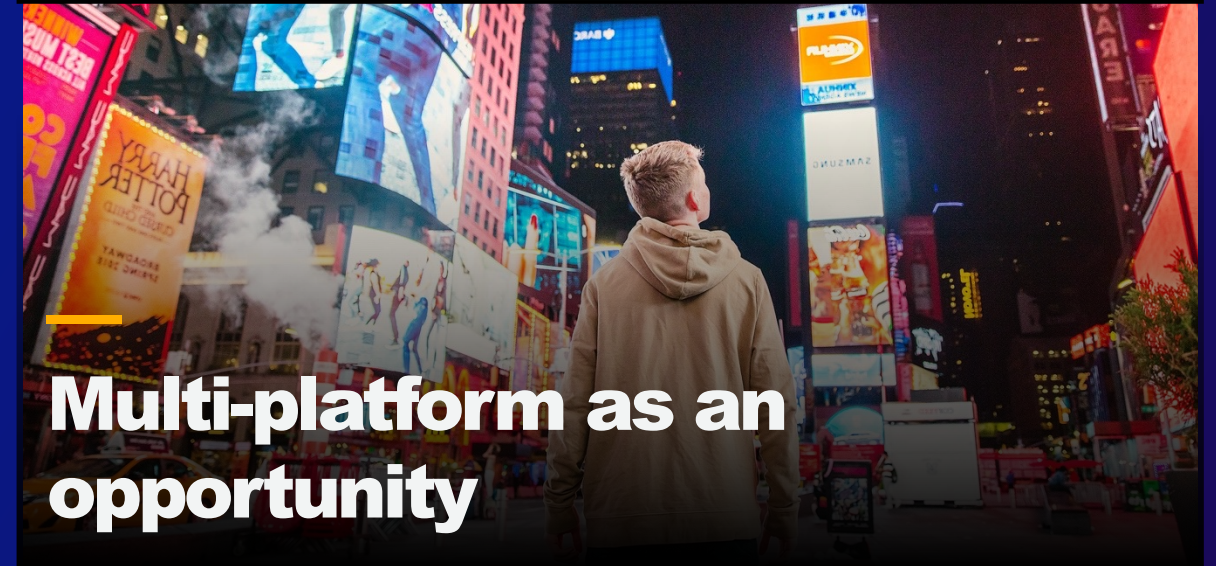
**Enhanced
Accuracy**

**Greater
Security**

**Delivered
Responsibly**



Re-imagining Digital Engineering with Amazon CodeWhisperer



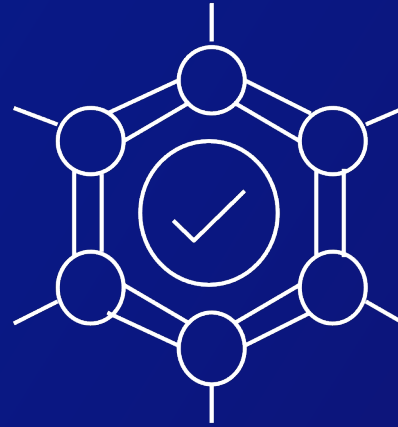
Top use cases



Accelerate
time-consuming
development tasks



Quickly adopt
new technologies
to build complex
solutions



Adopt AWS
services more
easily and with
confidence



Improve
application
security

Developer productivity with CodeWhisperer

57%

faster

27%

more likely to succeed

During the preview, Amazon ran a productivity challenge and participants who used CodeWhisperer were 27% more likely to complete tasks successfully and did so an average of 57% faster than those who did not use CodeWhisperer

CodeWhisperer is free to use for individual developers



Get started



**Dive deep with
a workshop**



**Learn how to build an
event-driven
serverless app**

Thank you!

Om Prakash Jha

[linkedin.com/in/omprakashjha](https://www.linkedin.com/in/omprakashjha)

Laura Al-Richane

[linkedin.com/in/laura-al-richane](https://www.linkedin.com/in/laura-al-richane)



Please complete the session survey in the mobile app

Pandurang Kamat

[linkedin.com/in/pandurang](https://www.linkedin.com/in/pandurang)