Experience In Boeing:

Created a CI pipeline in azure and integrated Coverity scans in pipeline.

Kraken

PS innovation

**Projects:**

Innovation Submission Platform for Boeing Employees:

The platform streamlines the process from idea generation to Quality Invention Disclosure (QID) approval, fostering a culture of innovation within the organization.

**Key Responsibilities:**

* Led the end-to-end development lifecycle of the platform, from conception to deployment.
* Design the RESTful APIs contract as well as cardinality for DB schema.
* Integrated JWT (JSON Web Tokens) for secure authentication and authorization of users.

**Impact**

* Reduced the turnaround time for idea evaluation and approval from 4 months to just 3 weeks, leading to a significant increase in patent and disclosures.
* **Technologies Used:**
* Spring Boot
* Java
* REST API
* JWT (JSON Web Tokens)
* MySQL

TANZU cloud provided by VMware . We have created an application PSinnovation (30GB)and created different dev space ,test space . Inside these space we have different application Like PSInnovationVenue\_MySQL, and psinnovations-api-dev

.

We login to the cloud foundary using cfcli command and connect to our space cf login -a https://api.system.taspre-phx.cloud.boeing.com –sso and then do cf push to push our manifest file. And it will deploy the instance to the application.

,

TALK TO HIM ABOUT CI/CD in angular

**MOST CHALLENGING WORK FACED:**

In our project we had an issue recently where a user was not able to save a DWR he created . Every DWR is associated with a GLOBAL FTWS (unique serial for an aircraft).

Once user lands on a page he can associate multiple FTWS number with multiple DWR at a time.

So once he landed on the page he received a next available FTWS number from an aircraft. His transaction begin and he keeps adding FTWS to DWR and it is auto incremented.

When he clicks on save he was not able to save as some other user are also trying to edit the same aircraft so he also received same FTWS number.

Earlier how we were handling it was on DB isolation. It was repeatable reads isolation for some of the transaction and we needed to change to serializable so that if a user is working on that next avail FTWS number **no other user is allowed to even read that**