

JENKINS SCREENSHOTS

The screenshot shows the Jenkins web interface for a job named 'inventory_mgmt'. The browser address bar displays 'http://localhost:8082/job/inventory_mgmt/'. The page has a sidebar on the left with navigation links: Status (selected), Changes, Build Now, Configure, Delete Pipeline, Stages, Rename, and Pipeline Syntax. The main content area shows the job's status as 'inventory_mgmt' with a green checkmark. Below this, there are 'Permalinks' for various build events: Last build (#3), Last stable build (#3), Last successful build (#3), Last failed build (#2), Last unsuccessful build (#2), and Last completed build (#3). A 'Builds' table on the left lists three builds: #3 (3:10 AM, green), #2 (2:56 AM, red), and #1 (2:38 AM, red). The bottom right corner indicates 'REST API' and 'Jenkins 2.528.3'.

The screenshot shows the Jenkins console output for the 'inventory_mgmt' job, build #3. The browser address bar displays 'http://localhost:8082/job/inventory_mgmt/3/console'. The console output shows a series of container startup messages for various services: inventory-mgmt-app-config-server-1, inventory-mgmt-app-api-gateway-1, inventory-mgmt-app-mysql-shared-1, inventory-mgmt-app-order-service-1, inventory-mgmt-app-inventory-service-1, inventory-mgmt-app-user-service-1, inventory-mgmt-app-billing-service-1, inventory-mgmt-app-rabbitmq-1, and inventory-mgmt-app-email-service-1. The output also shows pipeline execution steps: [Pipeline] }, [Pipeline] // dir, [Pipeline] }, [Pipeline] // stage, [Pipeline] }, [Pipeline] // withEnv, [Pipeline] }, [Pipeline] // node, and [Pipeline] End of Pipeline. The final status is 'Finished: SUCCESS'.

The screenshot shows the Jenkins configuration page for the 'inventory_mgmt' job. The browser address bar displays 'http://localhost:8082/job/inventory_mgmt/configure'. The page has a sidebar on the left with navigation links: General (selected), Triggers, Pipeline (selected), and Advanced. The main content area shows the 'Pipeline' configuration section. The 'Definition' dropdown is set to 'Pipeline script'. The 'Script' field contains a Groovy script that defines a pipeline with a 'stage' named 'Build' containing several 'steps' for building Docker images. The script is as follows:

```
9      git branch: 'main', url: 'https://github.com/isananya/inventory-mgmt.git'
10    }
11  }
12
13  stage('Build'){
14    steps(
15      dir('backend') {
16        bat "mvn -Dmaven.test.failure.ignore=true -DskipTests=true package"
17
18        bat "docker build -t api-gateway ./api-gateway"
19        bat "docker build -t billing-service ./billing-service"
20        bat "docker build -t order-service ./order-service"
21        bat "docker build -t config-server ./config-server"
22        bat "docker build -t email-service ./email-service"
23        bat "docker build -t eureka-server ./eureka-server"
24      }
25    )
26  }
```

At the bottom, there is a checkbox labeled 'Use Groovy Sandbox' which is checked. Below the script field are 'Save' and 'Apply' buttons.