



Journey Presentation

Harsh Thakur

Harsh.Thakur@shell.com



DAY-15

Python – Day 3

1. Pandas : Panel Datasets

- a. Dataframe
- b. Functions on dataframe
- c. Changing column to lowercase
- d. Dropping Null Values
- e. Dropping columns with Null Values
- f. Filling null values
- g. Describe columns
- h. Value counts()
- i. Selected rows or columns of dataframe
- j. Conditional selection

2. Matplotlib

3. Arrays In Python


```
File Edit View Insert Cell Kernel Help
Python 3 (ipykernel)

c:\Users\user> conda env create --name dataFrame
CondaEnv: 1000 entries, 1 to 1000
Data columns (total 11 columns):
# Column Non-Null Count Dtype
---
0 title 1000 non-null object
1 genre 1000 non-null object
2 director 1000 non-null object
3 director 1000 non-null object
4 director 1000 non-null object
5 year 1000 non-null int64
6 rating 1000 non-null float64
7 revenue_million 1000 non-null float64
8 metacritic 1000 non-null float64
9 metacritic 1000 non-null float64
dtype: object(11, dtype=object)
memory usage: 93.8+ KB

In [47]: finalDF.describe()
Out[47]:
```

	year	rating	revenue_million	metacritic
mean	1993.000000	1.00000000	1.000000e+02	100.000000
max	2012.000000	10.00000000	1.000000e+09	100.000000
min	1929.000000	0.00000000	1.000000e+00	0.00000000
std	3.000000	3.00000000	1.000000e+08	10.000000
0%	1929.000000	0.00000000	1.000000e+00	0.00000000
25%	1993.000000	1.00000000	1.000000e+02	100.000000
50%	2000.000000	5.00000000	1.000000e+06	50.000000
75%	2008.000000	8.00000000	1.000000e+08	80.000000
100%	2012.000000	10.00000000	1.000000e+09	100.000000

```
In [ ]:
```

```
File Edit View Insert Cell Kernel Help
Python 3 (ipykernel)

In [45]: finalDF.describe()
Out[45]:
```

	year	rating	revenue_million	metacritic
mean	1993.000000	1.00000000	1.000000e+02	100.000000
max	2012.000000	10.00000000	1.000000e+09	100.000000
min	1929.000000	0.00000000	1.000000e+00	0.00000000
std	3.000000	3.00000000	1.000000e+08	10.000000
0%	1929.000000	0.00000000	1.000000e+00	0.00000000
25%	1993.000000	1.00000000	1.000000e+02	100.000000
50%	2000.000000	5.00000000	1.000000e+06	50.000000
75%	2008.000000	8.00000000	1.000000e+08	80.000000
100%	2012.000000	10.00000000	1.000000e+09	100.000000

```
In [46]: finalDF[revenue_million].value_counts().head()
Out[46]:
```

revenue_million	count
1000000000	1
100000000	1
10000000	1
1000000	1
100000	1

```
In [47]: finalDF[revenue_million].value_counts().head()
Out[47]:
```

revenue_million	count
1000000000	1
100000000	1
10000000	1
1000000	1
100000	1

```
File Edit View Insert Cell Kernel Help
Python 3 (ipykernel)

In [43]: finalDF.describe()
Out[43]:
```

	year	rating	revenue_million	metacritic
mean	1993.000000	1.00000000	1.000000e+02	100.000000
max	2012.000000	10.00000000	1.000000e+09	100.000000
min	1929.000000	0.00000000	1.000000e+00	0.00000000
std	3.000000	3.00000000	1.000000e+08	10.000000
0%	1929.000000	0.00000000	1.000000e+00	0.00000000
25%	1993.000000	1.00000000	1.000000e+02	100.000000
50%	2000.000000	5.00000000	1.000000e+06	50.000000
75%	2008.000000	8.00000000	1.000000e+08	80.000000
100%	2012.000000	10.00000000	1.000000e+09	100.000000

```
In [44]: finalDF[revenue_million].value_counts().head()
Out[44]:
```

revenue_million	count
1000000000	1
100000000	1
10000000	1
1000000	1
100000	1

```
In [45]: finalDF[revenue_million].value_counts().head()
Out[45]:
```

revenue_million	count
1000000000	1
100000000	1
10000000	1
1000000	1
100000	1

```
File Edit View Insert Cell Kernel Help
Python 3 (ipykernel)

In [47]: finalDF[revenue_million].value_counts().head()
Out[47]:
```

revenue_million	count
1000000000	1
100000000	1
10000000	1
1000000	1
100000	1

```
In [48]: finalDF[revenue_million].value_counts().head()
Out[48]:
```

revenue_million	count
1000000000	1
100000000	1
10000000	1
1000000	1
100000	1

```
In [49]: finalDF[revenue_million].value_counts().head()
Out[49]:
```

revenue_million	count
1000000000	1
100000000	1
10000000	1
1000000	1
100000	1

```
File Edit View Insert Cell Kernel Help
Python 3 (ipykernel)

In [49]: finalDF[revenue_million].value_counts().head()
Out[49]:
```

revenue_million	count
1000000000	1
100000000	1
10000000	1
1000000	1
100000	1

```
In [50]: finalDF[revenue_million].value_counts().head()
Out[50]:
```

revenue_million	count
1000000000	1
100000000	1
10000000	1
1000000	1
100000	1

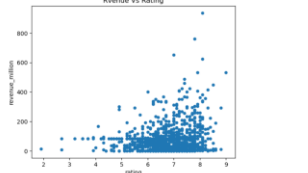
```
In [51]: finalDF[revenue_million].value_counts().head()
Out[51]:
```

revenue_million	count
1000000000	1
100000000	1
10000000	1
1000000	1
100000	1

```
File Edit View Insert Cell Kernel Help
Python 3 (ipykernel)

In [55]: import matplotlib.pyplot as plt
WARNING:matplotlib.font_manager.FontManager:building the font cache, this may take a moment.

In [57]: finalDF.plot(kind='scatter', x='rating', y='revenue_million', title='Revenue Vs Rating')
Out[57]:
```





DAY-16

Apache PySpark – Day 1

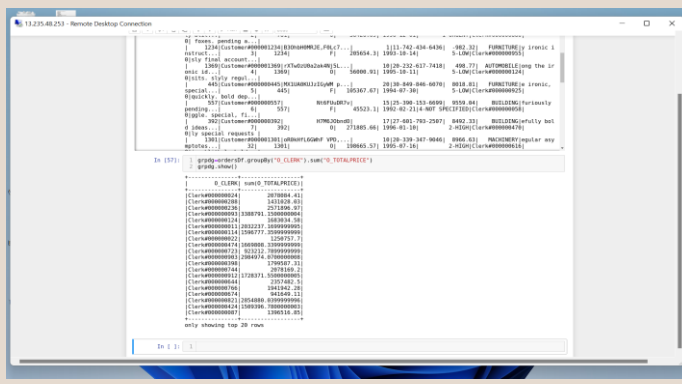
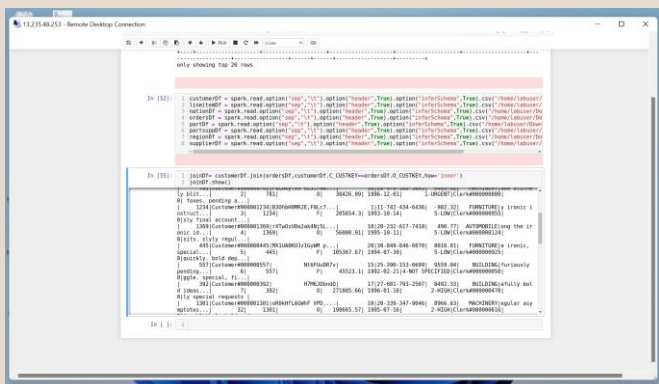
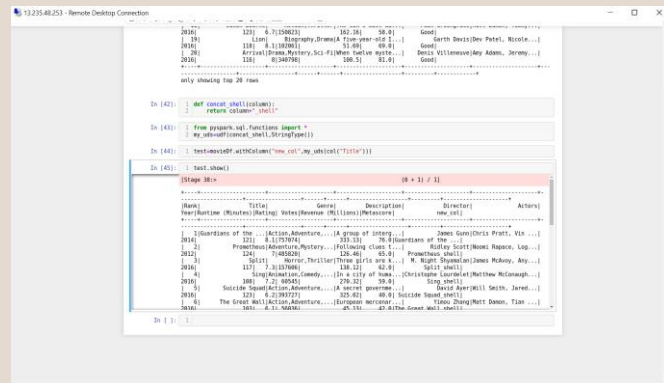
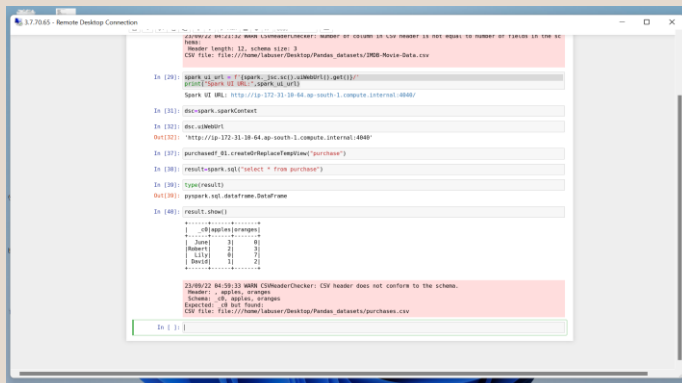
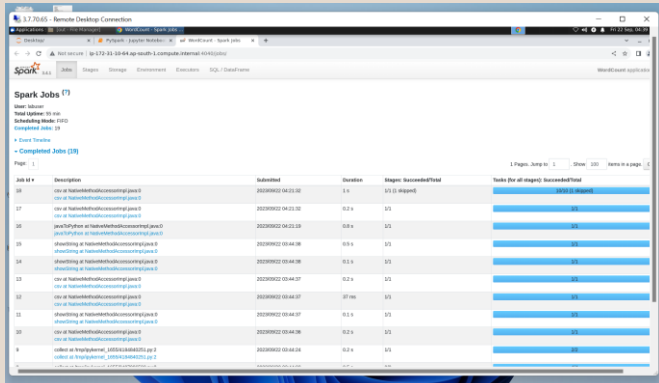
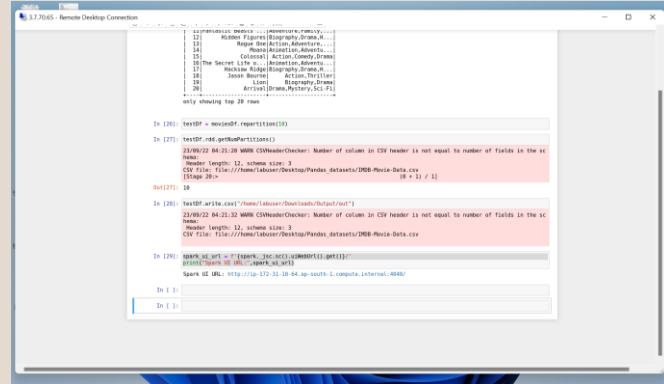
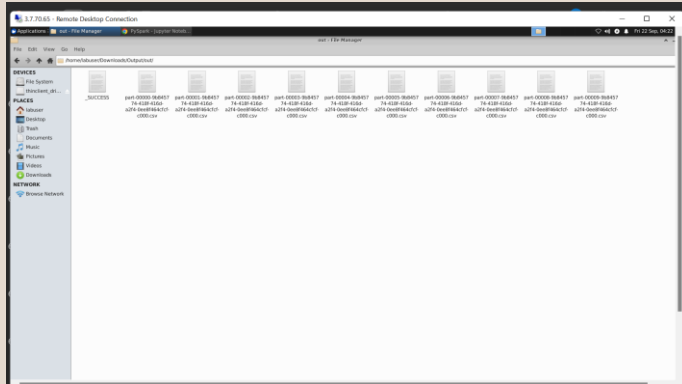
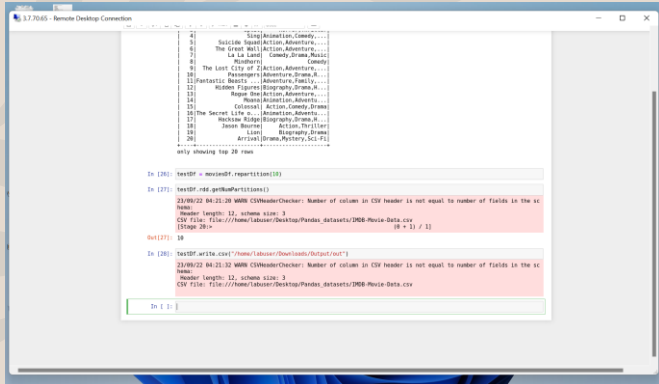
1. Driver process
2. Executors
3. Cluster Manager
4. Partition
5. Job, Stage and Task
6. Lazy Evaluation
7. RDD (Resilient Distributed Dataset)
8. Caching
9. Catalyst and Tungsten
10. RDD vs Dataframe
11. Deployment Types
 1. Client Mode
 2. Cluster Mode



DAY-17

Apache PySpark – Day 2

1. Partitioning
2. Spark UI URL
3. SQL in PySpark
4. Drop duplicates
5. When – Otherwise condition
6. Concat and Functions
7. Caching
8. Joins
9. Group By





DAY-18

Apache PySpark – Day 3

1. Caching
2. Persist and different storage levels
3. Aggregation
4. Managed and External Tables
5. Partitioning
6. For-Each

**** Note : Only 4 hours training due to OneIDA event. ****



DAY-19

Azure Databricks – Day 1

1. What is databricks?
2. Setting Up azure databricks
3. Overview of Databricks Workspace
4. Cluster configuration
5. Mounting
6. Widgets
7. Parameters
8. Functions
9. %run, %md, %language
10. Structured streaming

Microsoft Azure | Search resources, services, and docs (2/2)

AnushkaIDA_anushkadatabricks | Overview

Deployment

Search [x] Delete Cancel Refresh

Your deployment is complete

Deployment name: AnushkaIDA_anushkadatabricks
Subscription: resource/16934220094/rgpnu...
Resource group: AnushkaIDA

Start time: 9/26/2023, 9:42:41 AM
Completion ID: bc7f632-799c-4c2b-b64b-ed4368L

Cost management
Get notified to stay within your budget and prevent unexpected charges on your bill.
[Set up cost alerts >](#)

Microsoft Defender for Cloud
Secure your apps and infrastructure
[Go to Microsoft Defender for Cloud >](#)

Free Microsoft tutorials
[Start learning today >](#)

Work with an expert
Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support.
[Find an Azure expert >](#)

Give feedback
[Tell us about your experience with deployment](#)

Deployment details
Next steps
[Go to resource](#)

Microsoft Azure | databricks | Search data, notebooks, recent, and more...

Get started with Databricks

Start the SQL warehouse
A SQL warehouse is a simple compute resource that gives you the power to process data in the cloud. We've created your first warehouse for you. To start it, click the button below.
[Start warehouse](#)

Explore sample projects
Don't have your data handy? Learn how to gain insights in just a few steps with these sample projects.

SQL and visualizations
KPI dashboard
Data analysis in Python

Bring in your own data
Connect to over 50+ data sources and create tables in Databricks.
[Skip onboarding](#)

Microsoft Azure | databricks | Search data, notebooks, recent, and more...

Shellnext's Cluster

Configuration | Notebooks (0) | Libraries | Event log | Spark UI | Driver logs | Metrics | Apps | Spark compute UI - Master

Policy: Unrestricted

Access mode: Multi mode | Single mode
Single user: Shellnext

Performance
Databricks Runtime Version
13.3 LTS (includes Apache Spark 3.4.1, Scala 2.12)

Use Photon Acceleration

Worker type: Standard_DS1_v2
Min workers: 2
Max workers: 8
Current: 2
Spot instances: ☐

Driver type: Standard_DS1_v2
14 GB Memory, 4 Cores

Summary
2-8 Workers
28-32 GB Memory
8-32 Cores
1 Driver
14 GB Memory, 4 Cores
Runtime: 13.3 LTS (includes Apache Spark 3.4.1, Scala 2.12)
Photon | Standard_DS1_v2 | 4-14

Microsoft Azure | databricks | Search data, notebooks, recent, and more...

Mounting a file system

Run all | Shellnext's Cluster | Schedule | Stop

1 dbutils.fs.mount(source = "wasbs://input@anushka2.blob.core.windows.net",
mount_point = "/mnt/input",
extra_configs = {"fs.azure.account.key.anushka2.blob.core.windows.net":dbutils.secrets.get(scope = "anushkascope", key = "anushka-secret")})

2 dbutils.fs.mounts()

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

1000



DAY-20

Azure Databricks – Day 2

1. Structured streaming
2. DBFS from sample data
3. Delta table & Parquet table
4. Partition Tables
5. Medallion / Multi hop Architecture
6. Workflows → Jobs
7. Accessing data from Azure SQL database
8. Unity catalog in databricks
9. Metastore

[illegible]

The screenshot shows the Databricks Catalog Explorer interface. On the left sidebar, the 'Catalog' section is expanded, showing a tree view with 'hive_metastore' and 'default' (selected). Below this, there are sections for 'Viz', 'SQL Editor', 'Queries', 'Dashboards', 'Alerts', 'Query History', and 'SQL Worksheets'. The main panel displays the 'default.zipcodes_1_csv' table. It shows the owner as 'Not set', size as '3.0GB', and 1 file. Below this, there are tabs for 'Columns', 'Sample Data', 'Details', 'Permissions', and 'History'. The 'Columns' tab is active, showing a table with columns: 'RecordNumber' (int), 'Zipcode' (int), 'ZipCodeType' (string), 'City' (string), 'State' (string), 'LocationType' (string), 'Lat' (double), 'Long' (double), 'Xaxis' (double), and 'Yaxis' (double). Each column has a dropdown arrow next to it.

The screenshot shows the Microsoft Azure Databricks interface. At the top, there's a search bar and navigation tabs for 'New', 'Workspace', 'Catalog', 'Workflows', 'Compute', 'SQL Editor', 'Clusters', 'Alerts', 'Query History', and 'SQL Workflows'. The main area displays a notebook with a REST API call to the 'history' endpoint of a Spark job. Below the code, a table of results is shown, containing columns for version, timestamp, user_id, user_name, and operation. The table has two rows of data. The interface also shows a sidebar with navigation options and a bottom bar with various tool icons.

version	timestamp	user_id	user_name	operation
1	2023-09-27T04:18:15.000-0000	87347278920035	phillip@msn.com	CCP: INFO
2	2023-09-27T04:18:15.000-0000	87347278920035	phillip@msn.com	CREATE: TABLE

[illegible]

The screenshot shows the Databricks interface. At the top, there's a navigation bar with 'Microsoft Azure', 'databricks', and a search bar. Below it, a sidebar on the left contains icons for 'Workspace', 'Recent', 'Catalog', 'Workspace', and 'Compute'. The main area displays a notebook titled '27th September Python V1'. The notebook content includes a 'Run' button, a 'Table' view of the 'part' table, and a 'Table' view of the 'part' table. The 'part' table view shows columns like 'part_id', 'part_name', 'material', 'weight', 'length', 'width', 'height', 'volume', 'surface_area', 'unit_cost', and 'supplier_name'. The 'Table' view shows a single row with 10 columns: 'part_id', 'part_name', 'material', 'weight', 'length', 'width', 'height', 'volume', 'surface_area', and 'unit_cost'. The 'Table' view is currently showing 1 row out of 2,702 columns.

The screenshot shows the AWS IAM console interface. On the left, the 'Groups' page is selected. The main content area shows the 'AWS managed' group. The 'Permissions' tab is active, displaying a list of permissions. The 'Attach permissions' button is highlighted. The permissions list includes 'AmazonS3ReadOnlyAccess', 'AmazonEC2ReadOnlyAccess', and 'AmazonECSReadOnlyAccess'. The 'Attach permissions' button is located at the bottom right of the permissions list.

The screenshot displays the Databricks Jobs interface. On the left is a dark sidebar with navigation options: Home, Workspace, Recently, Clusters, Workspace, Clusters, Jobs, SQL Editor, Queries, Dashboards, Alerts, Query History, and SQL Workflows. The main panel is titled 'Workflows' and features a 'Free trial ends in 14 days. Upgrade to Premium to Run More Jobs' banner. Below the title, there's a section for 'Jobs' with a 'Job name' dropdown set to 'Delta Live Tables'. A filter 'Only my job runs' is selected. A timeline shows a job run on '2023-09-25 14:30:00' to '2023-09-27 14:30:00'. Below the timeline is a table of job runs:

Start time	Job	Run as	Launched	Duration	Status	Run parameters
Sep 27, 2023, 07:46 PM	Amolika Job	ⓘ DefaultUser	By scheduler	2s	ⓘ Skipped	
Sep 27, 2023, 07:46 PM	Amolika Job	ⓘ DefaultUser	By scheduler	2m-5s	ⓘ Pending	

At the bottom, there are links for 'View Logstream', 'Job Run', 'Data Inspection', and 'Delta Live Tables'. The footer includes 'Help & Learning' and 'Feedback'.

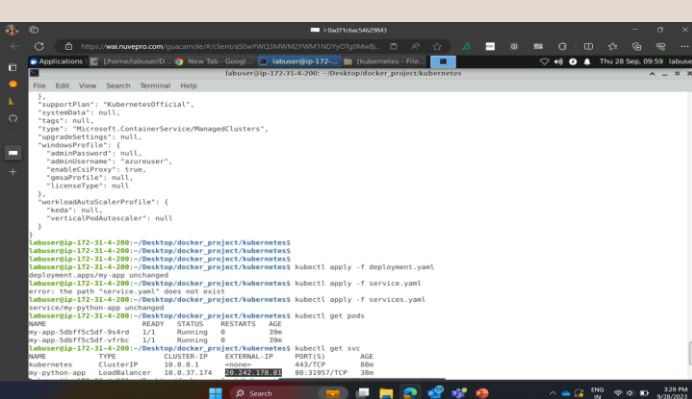
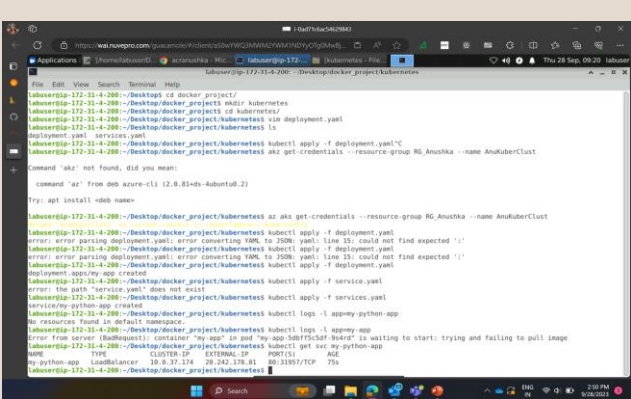
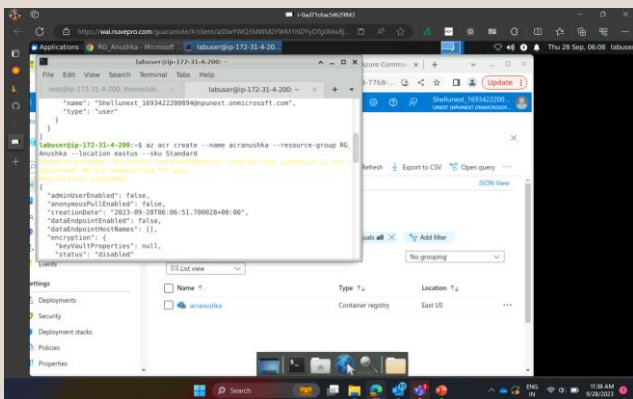
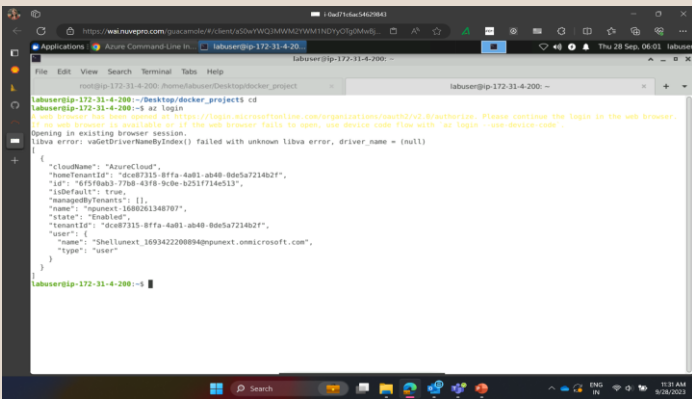
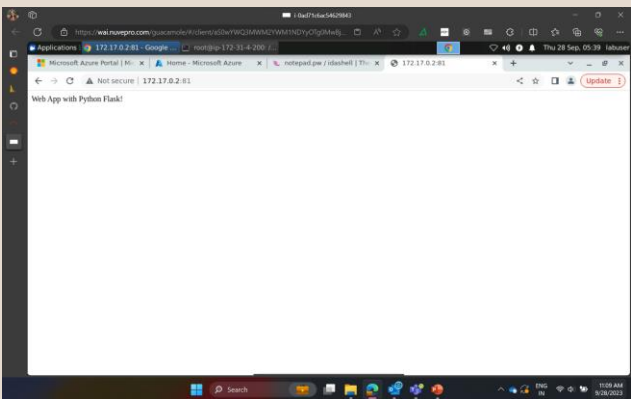
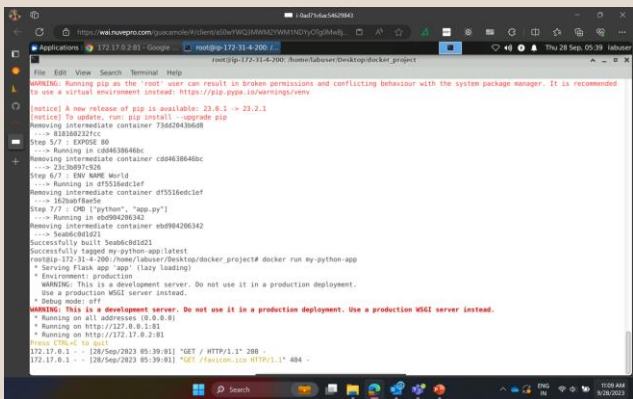
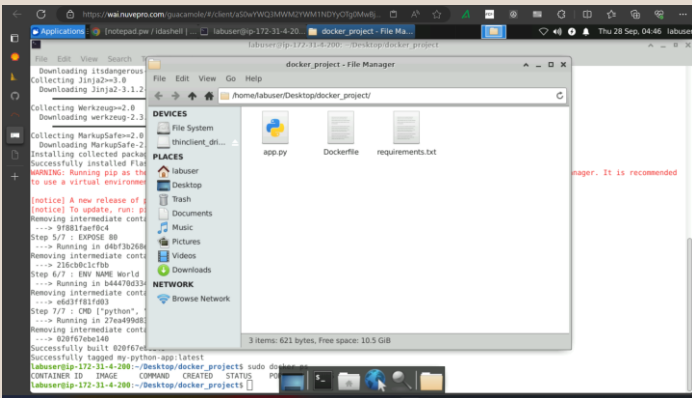
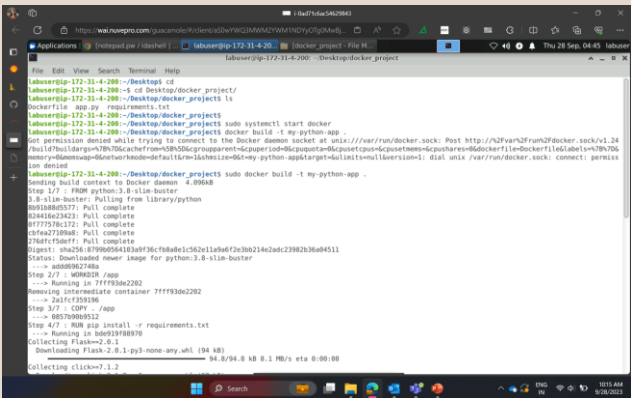
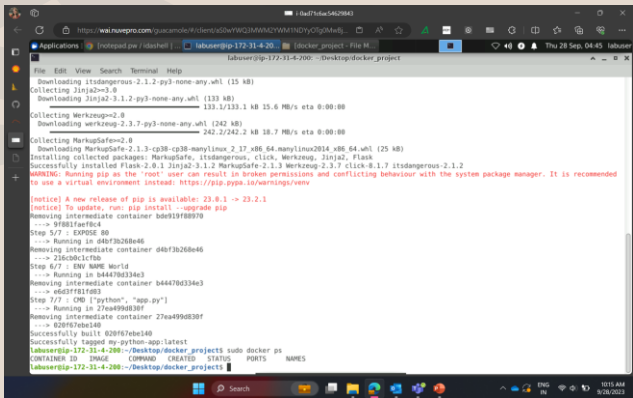
The screenshot shows the Microsoft Azure portal interface for configuring a workflow named 'Anushka-Job'. The left sidebar contains navigation options such as 'New', 'Workspace', 'Recent', 'Catalog', 'Workflows', 'Compute', 'SQL Editor', 'Queries', 'Dashboards', 'Alerts', 'Query History', 'SQL Warehouses', 'Data Engineering', 'Job Runs', 'Data Ingestion', 'Delta Live Tables', 'Machine Learning', 'Experiments', 'Feature Store', 'Monitoring', and 'Setup'. The main workspace area displays the 'Anushka-Job' workflow with a task named 'Anushka-Job_cluster' and an 'Add task' button. The right sidebar contains panels for 'Job details' (Job ID, Creator, Run as, Tags), 'Git' (Not configured), 'Schedule' (Every 2 minutes), and 'Compute' (Anushka-Job_cluster).



DAY-21

Docker Kubernetes

1. What is Docker ?
2. Docker initialization in VM
3. Build Docker image
 - Application
 - Requirement.txt
 - Dockerfile
4. Image created and Docker push
5. Azure container registry (ACR)
6. Kubernetes
7. AKS
8. Deploy a single-image application using code and UI



Microsoft Azure Portal | Microsoft Azure

Deploy a quickstart application

Resource	Type	Status
azure-vote	Namespace	Success
azure-vote-back	Deployment	Success
azure-vote-back	Service	Success
azure-vote-front	Deployment	Success
azure-vote-front	Service	Success

2. Next steps

Here are some actions you can take to further explore your new application and learn

[Previous](#) [Close](#)

Azure Voting App

Cats - 0 | Dogs - 0

DAY-22

Azure Deployment / DevOps

1. What is DevOps ?
 1. Collaborative Branch (Main branch)
 2. Feature branch
2. Azure DevOps
 1. Boards
 2. Repos
 3. Pipelines
 4. Test plans
3. Creating a project Board in Azure DevOps
 - Work items
 - Epic
 - Feature
 - User story
 - Tasks
4. Azure Repos
5. Azure Pipelines

Azure DevOps Shellunext1693422200894 / IDA_Anushka_Shell / Boards / Work Items

Try the New Boards Hub for improved performance, accessibility, and new features. [Click here to learn more.](#)

Work Items

Recently updated | + New Work Item | Open in Queries | Column Options | Import Work Items | Recycle Bin

Filter by keyword

ID	Title	Assigned To	Status	Area Path	Tags
24	Implement a search algorithm that queries the database for books	Unassigned	New	IDA_Anushka_Shell	
23	Design a search bar and results page UI	Unassigned	New	IDA_Anushka_Shell	
22	As a User, I want to search for books by title or author, so I ca...	Unassigned	New	IDA_Anushka_Shell	
21	Implement backend logic to fetch books by category	Unassigned	New	IDA_Anushka_Shell	
20	Develop a UI for browsing books by category	Unassigned	New	IDA_Anushka_Shell	
19	Create a database schema for book categories	Unassigned	New	IDA_Anushka_Shell	
18	As a user, I want to browse books by category, so I can find books ...	Unassigned	New	IDA_Anushka_Shell	
17	Search books	Unassigned	New	IDA_Anushka_Shell	
16	Browse books	Unassigned	New	IDA_Anushka_Shell	
15	Browse and Search Books	Unassigned	New	IDA_Anushka_Shell	
14	Handle authentication errors and provide appropriate feedback	Unassigned	New	IDA_Anushka_Shell	
13	Implement backend logic for user authentication	Unassigned	New	IDA_Anushka_Shell	
12	Design the login page UI	Unassigned	New	IDA_Anushka_Shell	
11	As a registered user, I want to log in to my account	Unassigned	New	IDA_Anushka_Shell	
10	User Login	Unassigned	New	IDA_Anushka_Shell	

Azure DevOps Shellunext1693422200894 / IDA_Anushka_Shell / Repos / Branches / IDA_Anushka_Shell

Branches

Mine All State

Search branch name

Branch	Comm...	Author	Authored ...	Behind / Ahead	Status	Pull ...
feature	2264f52	Shellunext unex...	52m ago	1 0		
main	8d3320c	Shellunext unex...	51m ago			

Azure DevOps Shellunext1693422200894 / IDA_Anushka_Shell / Repos / Pull requests / IDA_Anushka_Shell

Pull requests

Mine Active Completed Abandoned

New pull request

Pull Request ID	Created by	Assigned to	Target branch
updated test	Shellunext unex1D490 request 13 into feature		Completed 52m ago
Updated test:jon	Shellunext unex1D490 request 12 into feature		Completed 1h ago
updated test:jon	Shellunext unex1D490 request 11 into feature		Completed 1h ago

https://dev.azure.com/Shellunext1693422200894/IDA_Anushka_Shell/_git/IDA_Anushka_Shell/pullrequests?_a...

Azure DevOps Shellunext1693422200894 / IDA_Anushka_Shell / Pipelines / Anushka622001.sample-code / 20230629.1

Summary Tests

Triggered by Anushka622001

Repository and version: Anushka622001/sample-code, feature/main 4eb0b5e

Time started and elapsed: Just now, 1m 13s

Related: 0 work items, 0 artifacts

Tests and coverage: Get started

Errors: 4 Warnings: 2

failed to download Python from the GitHub Actions python registry (https://github.com/actions/python-versions). Error: Error: Could not find Python matching spec 2.7 (p64) in the python-ve... Job Python27 + Use Python 2.7

Version spec 2.7 for architecture x64 did not match any version in Agent.ToolsDirectory. Versions in /opt/hostedtoolcache: 3.10.13 (p64) 3.11.5 (p64) 3.7.17 (p64) 3.8.18 (p64) 3.9.18 (p64) If this ... Job Python27 + Use Python 2.7

failed to download Python from the GitHub Actions python registry (https://github.com/actions/python-versions). Error: Error: Could not find Python matching spec 3.6 (p64) in the python-ve... Job Python36 + Use Python 3.6

Version spec 3.6 for architecture x64 did not match any version in Agent.ToolsDirectory. Versions in /opt/hostedtoolcache: 3.10.13 (p64) 3.11.5 (p64) 3.7.17 (p64) 3.8.18 (p64) 3.9.18 (p64) If this ... Job Python36 + Use Python 3.6

Jobs	Name	Status	Duration
Job Python27	Job Python27	Failed	4s
Job Python35	Job Python35	Queued	
Job Python36	Job Python36	Failed	5s
Job Python37	Job Python37	Success	16s

The background features a light gray base with large, soft-edged organic shapes in muted red and olive green. A thin white line outlines a shape on the right. In the top left, there is a faint, stylized illustration of a leafy branch.

Thank You