**Azure Deployment Workflow - Step-by-Step Lab Guide**

**Lab Objective**

Learn to work with ARM vs Bicep exports, parameter files, linked IR tokens, build→release pipelines, smoke tests, and back-out plans.

**Step 1: ARM vs Bicep Export**

1. In the Azure Portal, navigate to a resource group.
2. Click **Export template** to download the ARM JSON.
3. Install Bicep CLI: az bicep install.
4. Convert ARM to Bicep: bicep decompile template.json.
5. Open the Bicep file and review its cleaner syntax.
6. Save both ARM and Bicep versions for reference.

**Checkpoint**: You should have both .json and .bicep versions of your template.

**Step 2: Parameter Files**

1. Create a .parameters.json file for your Bicep or ARM template.
2. Define environment-specific variables (e.g., location, SKU).
3. Ensure no secrets are included.
4. Store secrets in Azure Key Vault and reference them from the template.

**Checkpoint**: Your parameter file loads successfully during template deployment.

**Step 3: Linked IR Tokens**

1. In Azure Data Factory, set up a self-hosted or Azure IR.
2. Generate a Linked IR token from the source environment.
3. Store the token in Azure Key Vault.
4. In the target environment, use the token to link IRs securely.
5. Test data movement to confirm connectivity.

**Checkpoint**: Data Factory can use the linked IR without errors.

**Step 4: Build → Release Pipeline**

1. In Azure DevOps, create a new pipeline.
2. **Build stage**:
   * Add tasks for Bicep linting (bicep build --stdout to validate).
   * Package templates and parameter files into an artefact.
3. **Release stage**:
   * Deploy to a test environment.
   * Run smoke tests (see Step 5).
4. Add manual approval for production.

**Checkpoint**: Your pipeline runs successfully and deploys to the test environment.

**Step 5: Smoke Test**

1. After deployment, run a lightweight validation script:
   * Example: Check a web app's HTTP 200 response.
2. Integrate this script into your release pipeline as a stage.

**Checkpoint**: Smoke test passes before production approval.

**Step 6: Back-out Plan**

1. Document rollback steps:
   * Identify last known good deployment.
   * Use Azure deployment history or stored artefacts to redeploy.
2. Simulate a failed deployment in a test environment.
3. Practice rolling back to the previous version.

**Checkpoint**: Rollback is completed successfully within the set SLA.

**Completion Checklist**

* ARM and Bicep files exported.
* Parameter file created and tested.
* Linked IR token set up securely.
* Build→release pipeline configured.
* Smoke test automated.
* Back-out plan documented and tested.

**Case Study: Release ADF to Stage via Pipeline & Pass Smoke Tests**

**Background**

A data engineering team manages multiple Azure Data Factory (ADF) environments (dev, stage, prod). They use Git integration and CI/CD pipelines (Azure DevOps or GitHub Actions) to promote ADF ARM templates (or Bicep) through environments. The team needs a repeatable, auditable process to release ADF changes to **Stage** and validate the release using automated smoke tests before seeking approval for production.

**Objectives**

* Automate deployment of ADF artifacts (pipelines, datasets, linked services) to the Stage environment.
* Use parameter files to inject environment-specific values (storage accounts, IR names, secret references).
* Run smoke tests automatically after deployment to validate critical workflows.
* Provide clear rollback and remediation steps if smoke tests fail.

**Preconditions**

* ADF repository with collaboration branch (e.g., main or adf\_publish) and CI build artefacts (ARM or Bicep templates).
* Service principal or Managed Identity with contributor rights to the target resource group.
* Linked Integration Runtimes (IR) configured and test credentials/tokens available via Key Vault.
* A release pipeline in Azure DevOps (or GitHub Actions) that can accept artefacts and parameter files.

**Pipeline Design (High Level)**

1. **Trigger**: A successful merge into the release branch (main / adf\_publish) or manual release trigger.
2. **Build Stage**:
   * Export ADF ARM templates (or build Bicep) and package artefacts.
   * Run basic lint/validation of the ARM JSON / Bicep