3. Build and release pipelines

└ **3.6 Infrastructure as Code (IaC)**

└ **3.6.2 Implement desired state: Bicep, ARM, DSC**

---

1. What does “desired state” mean in Infrastructure as Code?
2. What are the core differences between Bicep and ARM templates?
3. How do you declare and deploy resources with Bicep?
4. How do you declare and deploy resources with ARM templates?
5. What is Desired State Configuration (DSC) and when should you use it?
6. How do you apply and monitor DSC on Azure virtual machines?
7. How can you use Bicep and ARM templates together?
8. What are best practices for parameterizing templates in Bicep and ARM?
9. How do you validate and test Bicep or ARM templates before deployment?
10. What methods can you use for secure secrets management with Bicep, ARM, and DSC?

---

1. **What does “desired state” mean in Infrastructure as Code?**

It refers to defining the exact configuration of infrastructure resources that should exist. Tools apply this state declaratively, ensuring the environment matches the intended configuration automatically.

---

2. **What are the core differences between Bicep and ARM templates?**

Bicep is a simpler, declarative DSL that transpiles to *ARM* JSON templates. *Bicep* improves readability and authoring, while *ARM* is JSON-based, more verbose, and harder to maintain.

---

3. **How do you declare and deploy resources with Bicep?**

Write a .bicep file with resource blocks. Deploy using:

```
az deployment group create --resource-group <rg> --template-file main.bicep
```

---

4. **How do you declare and deploy resources with ARM templates?**

Use a JSON file defining resources under resources array. Deploy using:

```
az deployment group create --resource-group <rg> --template-file main.json
```

---

5. **What is Desired State Configuration (DSC) and when should you use it?**

DSC is a *PowerShell*-based tool to configure and enforce VM state (e.g., software installed, services running). Use it when managing OS-level configurations, not just *Azure* resources.

---

6. **How do you apply and monitor DSC on Azure virtual machines?**

Use *Azure Automation DSC* or *Azure Automanage Machine Configuration*. Onboard VMs, assign DSC configurations, and monitor compliance from the *Azure* portal.

---

7. **How can you use Bicep and ARM templates together?**

Bicep transpiles to *ARM* JSON for deployment. You can reference existing *ARM* templates in *Bicep* using module blocks or include *Bicep* files inside *ARM* with linked templates.

---

8. **What are best practices for parameterizing templates in Bicep and ARM?**

- Use parameters for values like names, locations, and sizes.
- Provide default values where safe.
- Avoid hardcoding.
- Use parameters.json files for reusable input sets.

9. **How do you validate and test Bicep or ARM templates before deployment?**
    - Use `bicep build` and `bicep linter`, or `az deployment validate` to catch errors early.
    - Run test deployments in non-prod environments to ensure correctness.

---

11. **What methods can you use for secure secrets management with Bicep, ARM, and DSC?**
    - Store secrets in *Azure Key Vault*.
    - Reference them using `reference()` or `getSecret()` in templates or variables.
    - Never embed secrets directly in the code or parameter files.