**4. Security and compliance**

**└ 4.2 Permissions and access**

**└ 4.2.1 Azure DevOps permissions, security groups**

---

1. What are the default security groups in Azure DevOps?
2. How do permissions inheritance and explicit overrides work in Azure DevOps?
3. How do you assign and manage permissions at organization, project, and object level?
4. What is the difference between 'Allow', 'Deny', and 'Not set' for permissions?
5. How do you grant least privilege access in Azure DevOps?
6. How are access levels different from security group permissions?
7. What are best practices for configuring permissions in Azure DevOps?
8. How can you audit and review effective permissions for a user or group?
9. What steps are required to restrict branch permissions or pipeline execution?
10. How do you manage external access (e.g., Stakeholders, outside collaborators)?

---

**1. What are the default security groups in Azure DevOps?**

*Azure DevOps* default security groups include *Project Administrators, Contributors, Readers, Build Administrators, and Project Collection Administrators*. Each group grants a specific set of permissions appropriate for common roles.

---

**2. How do permissions inheritance and explicit overrides work in Azure DevOps?**

Permissions are inherited from higher-level containers (organization or project) to lower-level resources (repos, pipelines, etc). Explicitly setting a permission ('Allow' or 'Deny') at a lower level overrides inherited permissions.

---

**3. How do you assign and manage permissions at organization, project, and object level?**

Navigate to *Organization/Project settings > Security*. Assign users/groups to built-in or custom security groups, then set permissions at the organization, project, or resource (repo, pipeline) level.

---

**4. What is the difference between 'Allow', 'Deny', and 'Not set' for permissions?**

'Allow' grants the permission, 'Deny' blocks it (overrides all 'Allow'), and 'Not set' means the permission is neither granted nor denied, so inheritance applies.

---

**5. How do you grant least privilege access in Azure DevOps?**

Add users to the group with the minimal permissions needed. Avoid granting *Project Administrator* unless required, and restrict direct assignments.

---

**6. How are access levels different from security group permissions?**

- Access levels (Stakeholder, Basic, Visual Studio, etc.) control which features are available.
- Security group permissions define what users can do within those features.

---

**7. What are best practices for configuring permissions in Azure DevOps?**

Use built-in groups for common roles, apply 'Deny' sparingly, review permissions regularly, and use custom groups only when necessary.

---

**8. How can you audit and review effective permissions for a user or group?**

Use the Permissions tab under the security settings of a resource. Select a user/group to view their effective permissions, including inherited and explicitly set rights.

**9. What steps are required to restrict branch permissions or pipeline execution?**
Go to the repository or pipeline, select Security, and set branch or pipeline permissions for specific users/groups (e.g., restrict force push or approve builds).

---

**10. How do you manage external access (e.g., Stakeholders, outside collaborators)?**
Assign external users to Stakeholder access or limit to specific security groups. Review permissions to ensure no unnecessary access is granted beyond required scope.