Azure Authentication for Pipelines

Service Principals & App Registrations

Pre-empting some pain

You want to run `az login` and `terraform init` in your GitHub Actions pipeline.

But pipelines are not users, they can't:

- Type in usernames and passwords (they can, but more on this next slide)
- Complete multi-factor authentication
- Use your personal Azure credentials

Solution: Service Principals

Why shouldn't I just give it my email and password?

What is a Service Principal?

Think of it as a **non-human user account** for applications and services.

It's essentially:

- An identity that applications can use to authenticate to Azure
- Has its own credentials (client ID, client secret, or certificate)
- Can be granted specific permissions to Azure resources (RBAC)
- Never expires unless you set it to

Service Principal vs App Registration

App Registration:

- The "blueprint" or definition of your application
- Lives in Azure Active Directory (Entra ID)
- Defines what your app is and what it can do

Service Principal:

- The "instance" of that app in a specific tenant
- The actual identity that gets permissions
- Created automatically when you register an app

Where Do They Live?

Azure Active Directory (Entra ID)

Navigate to: Azure Portal → Azure Active Directory → App registrations

What you'll find:

- Client ID (Application ID) - public identifier
- Tenant ID - your Azure AD directory ID
- Client secrets - the "passwords" for authentication
- API permissions - what this principal can access

Why Do We Need Them for Pipelines?

Authentication without human interaction:

```
# Instead of this (won't work in pipeline):
az login

# We can do this:
az login --service-principal \
  --username $CLIENT_ID \
  --password $CLIENT_SECRET \
  --tenant $TENANT_ID
```

You can do this locally to log in "as" a Service Principal, if you want to test access or similar.

Please be extra careful with these credentials.

No committing to GitHub

Avoid storing locally for longer than you need

They give someone access into our Azure Tenant, which we do not want!

Required Information for Pipeline (probably)

You'll need these four "bootstrapping" values to avoid a chicken&egg problem:

Variable	Description	Where to find
`CLIENT_ID`	Application ID	App registration overview
`CLIENT_SECRET`	The secret password	Certificates & secrets
`TENANT_ID`	Your Azure AD directory	App registration overview
`SUBSCRIPTION_ID`	Your Azure subscription	Subscriptions blade

Store these as GitHub secrets for now, never commit them to code

Once we have these, all other secrets should live in Azure Key Vaults, that we can access once we're logged in.

Permissions and Security

Best Practices:

- Grant minimum required permissions
- Use specific resource scopes, not subscription-wide when possible
- Rotate client secrets regularly
- Monitor service principal usage

Common roles for infrastructure:

- **Contributor:** Can create/modify resources
- **Reader:** Can only view resources
- **Custom roles:** Tailored to specific needs

What Happens in Your Pipeline

GitHub Actions authentication flow:

1. Pipeline starts and needs Azure access
2. Retrieves service principal credentials from secrets
3. Authenticates to Azure using these credentials
4. Can now run `az` commands and Terraform
5. Performs infrastructure operations with granted permissions

No human interaction required

Common Issues

Authentication failures:

- Check client secret hasn't expired
- Verify tenant ID is correct
- Ensure service principal has required permissions

Permission denied:

- Service principal needs proper role assignments
- Check scope of permissions (subscription vs resource group)
- Some operations require specific API permissions

Key Takeaways

- Service principals enable non-interactive Azure authentication
- Essential for CI/CD pipelines and automation
- Store credentials securely as secrets, never in code
- Grant minimum required permissions
- They live in Azure Active Directory (Entra ID)
- You'll use them for both Azure CLI and Terraform authentication

Next: Implement this in your CD pipeline for automated deployments