

Azure Cache for Redis - Environment Configurations

Development Environment

Local Development (No Azure)

```
bash

# .env.development.local
REDIS_URL=redis://localhost:6379/0
```

Azure Cache Basic Tier (Cheapest)

```
bash

# .env.development.azure
REDIS_URL=rediss://default:YourKey@dev-cache.redis.cache.windows.net:6380/0

# If SSL certificate issues during development
REDIS_SSL_CERT_REQS=none # NEVER use in production!
```

Azure Portal Settings:

- SKU: Basic C0 (250 MB) - ~\$16/month
- Non-SSL port: Disabled
- Minimum TLS version: 1.2
- Firewall: Add your development machine IP

Staging Environment

```
bash

# .env.staging
REDIS_URL=rediss://default:YourKey@staging-cache.redis.cache.windows.net:6380/0

# Explicit SSL settings
REDIS_SSL=true
REDIS_SSL_CERT_REQS=required
```

Azure Portal Settings:

- SKU: Standard C1 (1 GB) - ~\$61/month

- Features: Replication enabled
- Backup: Daily backup at 3 AM
- Firewall: Add staging server IPs
- Monitoring: Enable diagnostics

Production Environment

Option 1: Standard Tier (Most Common)

```
bash

# .env.production
REDIS_URL=rediss://default:YourKey@prod-cache.redis.cache.windows.net:6380/0

# Additional production settings
SECURE_COOKIES=true
COOKIE_DOMAIN=.yourdomain.com
RATE_LIMIT_DEFAULT=100 per minute # Stricter limits
```

Azure Portal Settings:

- SKU: Standard C3 (6 GB) - ~\$340/month
- High Availability: Enabled
- Persistence: Not available in Standard

Option 2: Premium Tier (High Scale)

```
bash

# .env.production.premium
REDIS_URL=rediss://default:YourKey@prod-premium.redis.cache.windows.net:6380/0

# With clustering (multiple connection strings)
REDIS_CLUSTER_NODES=["node1.redis.cache.windows.net:6380","node2.redis.cache.windows.net:6380"]
```

Azure Portal Settings:

- SKU: Premium P1 (6 GB) - ~\$555/month
- Features:
 - Data persistence (RDB/AOF)
 - Virtual Network support

- Clustering (up to 10 shards)
- Geo-replication
- Import/Export

Enterprise Production (Maximum Security)

With Private Endpoint

```
bash

# .env.production.enterprise
# Private endpoint - no public internet access
REDIS_URL=rediss://default:YourKey@prod-cache.privatelink.redis.cache.windows.net:6380/0

# Azure Key Vault for secrets
AZURE_KEY_VAULT_URL=https://your-keyvault.vault.azure.net/
```

With Managed Identity

```
python

# In your config.py
from azure.identity import DefaultAzureCredential
from azure.keyvault.secrets import SecretClient

# Get Redis password from Key Vault
credential = DefaultAzureCredential()
client = SecretClient(vault_url=settings.azure_key_vault_url, credential=credential)
redis_password = client.get_secret("redis-password").value
```

Sizing Guide

Environment	Users	Requests/sec	Recommended SKU	Cost/month
Dev/Test	<10	<100	Basic C0 (250MB)	~\$16
Small Prod	<100	<500	Standard C1 (1GB)	~\$61
Medium Prod	<1000	<2000	Standard C2 (2.5GB)	~\$152
Large Prod	<5000	<5000	Standard C3 (6GB)	~\$340
Enterprise	>5000	>5000	Premium P1+	>\$555

Deployment Checklist

Development → Staging

- ☐ Create staging Redis instance
- ☐ Configure firewall rules
- ☐ Enable monitoring
- ☐ Test with load testing tool
- ☐ Set up alerts

Staging → Production

- ☐ Create production Redis instance
- ☐ Enable high availability
- ☐ Configure backup schedule
- ☐ Set up geo-replication (Premium)
- ☐ Configure Private Endpoint (optional)
- ☐ Enable diagnostic logs
- ☐ Set up alerts for:
 - High memory usage (>80%)
 - High CPU usage (>80%)
 - Connection errors
 - Evicted keys
- ☐ Document connection strings in Key Vault
- ☐ Test failover scenarios



Important Security Notes

1. Never commit Redis keys to source control

```
bash
# .gitignore
.env
.env.*
!.env.example
```

2. Rotate access keys regularly

```
bash
# Azure CLI
az redis regenerate-keys --name prod-cache --resource-group rg-prod --key-type Primary
```

3. Use least privilege

- Production apps: Read/Write only
- Monitoring: Read only
- Admin: Full access (limited users)

4. Enable Redis ACL (Redis 6+)

```
redis
```

```
ACL SETUSER app-user on ~* &* -@dangerous +@read +@write
```

Migration Path

Moving from Basic → Standard

```
bash
```

```
# No downtime migration
```

1. Export data: `redis-cli --rdb backup.rdb`
2. Create new Standard instance
3. Import data via Portal
4. Update connection string
5. Test thoroughly
6. Delete old instance

Moving to Premium (with persistence)

```
bash
```

```
# Azure CLI
```

```
az redis import --name premium-cache --resource-group rg-prod \  
  --files "https://storage.blob.core.windows.net/backup/dump.rdb"
```

Monitoring Queries

Azure Monitor (KQL)

```
kql
```

```
// High memory usage
AzureMetrics
| where ResourceProvider == "MICROSOFT.CACHE"
| where MetricName == "UsedMemoryPercentage"
| where Average > 80
| summarize avg(Average) by bin(TimeGenerated, 5m), Resource
```

```
// Cache misses
AzureMetrics
| where MetricName == "CacheMisses"
| summarize sum(Total) by bin(TimeGenerated, 1h)
```

Redis CLI Monitoring

```
bash
```

```
# Real-time monitoring
```

```
redis-cli -h your-cache.redis.cache.windows.net -p 6380 -a YourKey --tls monitor
```

```
# Get slow queries
```

```
redis-cli -h your-cache.redis.cache.windows.net -p 6380 -a YourKey --tls slowlog get 10
```