

[**ArgoCD CLI Commands Cheat Sheet 3**](#_eedff4x8iyxf)

[📦 Application Management 3](#_u1rmnzppig0j)

[🔄 Sync Operations 4](#_touf8upahsux)

[🔙 Rollback & History 5](#_dfahw9hdtmi2)

[🔍 Monitoring & Debugging 5](#_7prnebpqg6ep)

[🎯 Project Management 6](#_uv7xopz90jdn)

[🌐 Cluster Management 7](#_3s9gkiosayfo)

[🔑 Repository Management 7](#_kvrcx27zsza6)

[🛠️ Admin Commands 8](#_wnrb6lwhhjng)

[📝 Common Workflows 8](#_x7b4u34uf7i)

[Deploy a new application 8](#_x6fu1y5798pk)

[Check application status and sync 9](#_abbpndson8dg)

[Troubleshoot sync issues 9](#_q0o8iuzbrcwd)

[Quick rollback 9](#_ho4e8znot6zx)

[🔗 Quick Tips 9](#_55jrahubrweq)

[Tab Completion 9](#_sd5tw8dhz29d)

[Environment Variables 9](#_wbx09w7i6lmd)

[Watch Mode 10](#_12tasik1hc6l)

[JSON Output 10](#_g3xp1sa6on6l)

# **ArgoCD CLI Commands Cheat Sheet**

The commands in this cheat sheet are compatible with ArgoCD v3.x.

**🔐 Authentication & Setup**

| # Login to ArgoCD server argocd login <SERVER\_ADDRESS> \  --username <USERNAME> \  --password <PASSWORD>  # Login with SSO argocd login <SERVER\_ADDRESS> --sso  # Get initial admin password argocd admin initial-password -n argocd  # Change your password argocd account update-password \  --current-password <OLD> \  --new-password <NEW>  # Logout argocd logout <SERVER\_ADDRESS>  # Re-authenticate current session argocd relogin  # Check CLI and server version argocd version  # Check your permissions argocd account can-i <ACTION> <RESOURCE> <PROJECT> # Example: argocd account can-i sync applications myproject |
| --- |

## **📦 Application Management**

| # List all applications argocd app list argocd app list -o yaml # Output in YAML format  # Get application details argocd app get <APP\_NAME> argocd app get <APP\_NAME> --refresh # Force refresh from Git  # Create a new application argocd app create <APP\_NAME> \  --repo <GIT\_REPO\_URL> \  --path <PATH\_IN\_REPO> \  --dest-server https://kubernetes.default.svc \  --dest-namespace <NAMESPACE> \  --project <PROJECT\_NAME>  # Create app from YAML file argocd app create -f app.yaml  # Update application source argocd app set <APP\_NAME> \  --repo <NEW\_REPO\_URL> \  --path <NEW\_PATH>  # Set application to auto-sync argocd app set <APP\_NAME> \  --sync-policy automated \  --auto-prune \  --self-heal  # Disable auto-sync argocd app set <APP\_NAME> --sync-policy none  # Delete an application argocd app delete <APP\_NAME> argocd app delete <APP\_NAME> --cascade # Also delete resources in cluster |
| --- |

## **🔄 Sync Operations**

| # Sync (deploy) an application argocd app sync <APP\_NAME>  # Sync with prune (remove deleted resources) argocd app sync <APP\_NAME> --prune  # Force sync (ignore validation errors) argocd app sync <APP\_NAME> --force  # Sync only specific resources argocd app sync <APP\_NAME> \  --resource <GROUP>:<KIND>:<NAMESPACE>/<NAME>  # Sync with specific revision (Git commit/tag/branch) argocd app sync <APP\_NAME> --revision <COMMIT\_SHA>  # Dry-run sync (preview changes) argocd app sync <APP\_NAME> --dry-run  # Wait for sync to complete argocd app wait <APP\_NAME> --timeout 300  # Terminate running sync argocd app terminate-op <APP\_NAME> |
| --- |

## **🔙 Rollback & History**

| # Show deployment history argocd app history <APP\_NAME>  # Rollback to previous version argocd app rollback <APP\_NAME>  # Rollback to specific history ID argocd app rollback <APP\_NAME> <HISTORY\_ID> |
| --- |

## **🔍 Monitoring & Debugging**

| # Show application status argocd app get <APP\_NAME> --show-params argocd app get <APP\_NAME> --show-operation  # Show diff between Git and cluster argocd app diff <APP\_NAME> argocd app diff <APP\_NAME> --local <LOCAL\_PATH> # Compare with local files  # View application logs argocd app logs <APP\_NAME> argocd app logs <APP\_NAME> --follow # Tail logs argocd app logs <APP\_NAME> --container <CONTAINER\_NAME> # Specific container  # Get application manifests argocd app manifests <APP\_NAME>  # List application resources argocd app resources <APP\_NAME>  # Show detailed resource tree argocd app get <APP\_NAME> --show-tree  # Refresh application (force reconciliation) argocd app get <APP\_NAME> --refresh argocd app get <APP\_NAME> --hard-refresh # Bypass cache |
| --- |

## **🎯 Project Management**

| # List all projects argocd proj list  # Get project details argocd proj get <PROJECT\_NAME>  # Create a new project argocd proj create <PROJECT\_NAME> \  --description "Project description" \  --dest <CLUSTER>,<NAMESPACE> \  --src <GIT\_REPO\_URL>  # Add allowed Git repository to project argocd proj add-source <PROJECT\_NAME> <GIT\_REPO\_URL>  # Add destination cluster to project argocd proj add-destination <PROJECT\_NAME> \  <CLUSTER\_URL> <NAMESPACE>  # Delete a project argocd proj delete <PROJECT\_NAME>  # Edit project RBAC policies argocd proj role list <PROJECT\_NAME> argocd proj role create <PROJECT\_NAME> <ROLE\_NAME> |
| --- |

## **🌐 Cluster Management**

| # List connected clusters argocd cluster list  # Add a cluster using kubeconfig context argocd cluster add <CONTEXT\_NAME> argocd cluster add <CONTEXT\_NAME> --name <CLUSTER\_NAME>  # Get cluster information argocd cluster get <CLUSTER\_NAME>  # Remove a cluster argocd cluster rm <CLUSTER\_NAME>  # Rotate cluster credentials argocd cluster rotate-auth <CLUSTER\_NAME> |
| --- |

## **🔑 Repository Management**

| # List configured repositories argocd repo list  # Add Git repository argocd repo add <GIT\_REPO\_URL> \  --username <USERNAME> \  --password <PASSWORD>  # Add Git repository with SSH key argocd repo add <GIT\_REPO\_URL> \  --ssh-private-key-path <PATH\_TO\_KEY>  # Add Helm repository argocd repo add <HELM\_REPO\_URL> \  --type helm \  --name <REPO\_NAME> \  --username <USERNAME> \  --password <PASSWORD>  # Remove repository argocd repo rm <REPO\_URL>  # Get repository details argocd repo get <REPO\_URL> |
| --- |

## **🛠️ Admin Commands**

| # Launch local dashboard (no cluster exposure) argocd admin dashboard argocd admin dashboard --port 8080  # Export all applications argocd admin export > backup.yaml  # Import applications argocd admin import -f backup.yaml  # Validate settings argocd admin settings validate  # Generate declarative config argocd admin export > argocd-backup.yaml  # Core mode (direct K8s API access, no login needed) argocd app list --core --kube-context <CONTEXT> |
| --- |

## **📝 Common Workflows**

### **Deploy a new application**

| argocd app create myapp \  --repo https://github.com/myorg/myrepo \  --path manifests \  --dest-server https://kubernetes.default.svc \  --dest-namespace production \  --sync-policy automated \  --auto-prune |
| --- |

### 

### **Check application status and sync**

| argocd app get myapp argocd app sync myapp --prune argocd app wait myapp --timeout 300 |
| --- |

### **Troubleshoot sync issues**

| argocd app diff myapp # Check differences argocd app get myapp --show-operation # Check sync operation argocd app logs myapp --follow # View logs |
| --- |

### **Quick rollback**

| argocd app history myapp # Get history ID argocd app rollback myapp 5 # Rollback to ID 5 |
| --- |

## **🔗 Quick Tips**

### Tab Completion

Enable bash/zsh completion for faster typing

| source <(argocd completion bash) |
| --- |

### Environment Variables

Set common flags as env vars

| export ARGOCD\_SERVER=argocd.example.com export ARGOCD\_AUTH\_TOKEN=<your-token> |
| --- |

### Watch Mode

Monitor app status continuously

| watch -n 2 argocd app get myapp |
| --- |

### JSON Output

Parse output with jq for automation

| argocd app list -o json | jq '.[] | select(.status.sync.status=="OutOfSync")' |
| --- |