Zarf CLI Docs

`zarf`

DevSecOps for Airgap

Synopsis

Zarf eliminates the complexity of air gap software delivery for Kubernetes clusters and cloud native workloads
using a declarative packaging strategy to support DevSecOps in offline and semi-connected environments.

```
zarf COMMAND [flags]
```

Options

- - a, –architecture
  `Architecture for OCI images and Zarf packages`
- - h, –help
  `help for zarf (default: false)`
- --insecure
  `Allow access to insecure registries and disable other recommended security enforcements such as package checksum and signature validation. This flag should only be used if you have a specific reason and accept the reduced security posture. (default: false)`
- - l, –log-level
  `Log level when running Zarf. Valid options are: warn, info, debug, trace (default: info)`
- --no-log-file
  `Disable log file creation (default: false)`
- --no-progress
  `Disable fancy UI progress bars, spinners, logos, etc (default: false)`
- --tmpdir
  `Specify the temporary directory to use for intermediate files`
- --zarf-cache
  `Specify the location of the Zarf cache directory (default: ~/.zarf-cache)`

`zarf bundle`

Zarf commands for creating, deploying, removing, pulling, and inspecting bundles

Options

- - h, –help
  `help for bundle (default: false)`
- --oci-concurrency
  `Number of concurrent layer operations to perform when interacting with a remote bundle. (default: 3)`

`zarf bundle create`

Create a Zarf bundle from a given directory or the current directory

Synopsis

```
zarf bundle create [DIRECTORY] [flags]
```

Options

- - c, –confirm
  `REQUIRED. Confirm the removal action to prevent accidental deletions (default: false)`
- - h, –help
  `help for create (default: false)`
- - o, –output
  `Specify the output (an oci:// URL) for the created Zarf bundle`
- - s, –set
  `Specify bundle template variables to set on the command line (KEY=value) (default: [])`
- - k, –signing-key
  `Path to private key file for signing bundles`
- - p, –signing-key-password
  `Password to the private key file used for signing bundles`

`zarf bundle deploy`

Deploy a Zarf bundle from a local file or URL (runs offline)

Synopsis

```
zarf bundle deploy [BUNDLE] [flags]
```

Options

- - h, –help
  `help for deploy (default: false)`
- - p, –packages
  `Specify the package(s) to deploy from the bundle.  E.g. packages=cluster-init,bigbang (default: [])`
- - s, –set
  `Specify deployment variables to set on the command line (KEY=value) (default: [])`

`zarf bundle inspect`

Display the zarf.yaml of a compiled Zarf bundle (runs offline)

Synopsis

```
zarf bundle inspect [BUNDLE] [flags]
```

Options

- - h, –help
  `help for inspect (default: false)`
- - k, –key
  `Path to a public key file that will be used to validate a signed bundle`

`zarf bundle pull`

Pull a Zarf bundle from a remote reigstry and save to the local file system

Synopsis

```
zarf bundle pull [OCI_REF] [flags]
```

Options

- - h, –help
  `help for pull (default: false)`
- - k, –key
  `Path to a public key file that will be used to validate a signed bundle`
- - o, –output
  `Specify the output directory for the pulled Zarf bundle`

`zarf bundle remove`

Remove a Zarf bundle or sub-packages that have been deployed already

Synopsis

```
zarf bundle remove [BUNDLE_NAME|BUNDLE_TARBALL|OCI_REF] [flags]
```

Options

- - c, –confirm
  ```
  REQUIRED. Confirm the removal action to prevent accidental deletions (default: false)
  ```
- - h, –help
  ```
  help for remove (default: false)
  ```
- - p, –packages
  ```
  Specify the package(s) to remove from the bundle.  E.g. packages=cluster-init,bigbang
  (default: [])
  ```

`zarf connect`

Accesses services or pods deployed in the cluster

Synopsis

Uses a k8s port-forward to connect to resources within the cluster referenced by your kube-context. Three default options for this command are <REGISTRY|LOGGING|GIT>. These will connect to the Zarf created resources (assuming they were selected when performing the `zarf init` command).

Packages can provide service manifests that define their own shortcut connection options. These options will be printed to the terminal when the package finishes deploying.
If you don't remember what connection shortcuts your deployed package offers, you can search your cluster for services that have the 'zarf.dev/connect-name' label. The value of that label is the name you will pass into the 'zarf connect' command.

Even if the packages you deploy don't define their own shortcut connection options, you can use the command flags to connect into specific resources. You can read the command flag descriptions below to get a better idea how to connect to whatever resource you are trying to connect to.

```
zarf connect { REGISTRY | LOGGING | GIT | connect-name } [flags]
```

Options

- --cli-only
  `Disable browser auto-open (default: false)`
- - h, –help
  `help for connect (default: false)`
- --local-port
  `(Optional, autogenerated if not provided) Specify the local port to bind to.  E.g. local-port=42000 (default: 0)`
- --name
  `Specify the resource name.  E.g. name=unicorns or name=unicorn-pod-7448499f4d-b5bk6`
- --namespace
  `Specify the namespace.  E.g. namespace=default (default: zarf)`
- --remote-port
  `Specify the remote port of the resource to bind to.  E.g. remote-port=8080 (default: 0)`
- --type
  `Specify the resource type.  E.g. type=svc or type=pod (default: svc)`

`zarf connect list`

Lists all available connection shortcuts

Synopsis

```
zarf connect list [flags]
```

Options

- - h, –help
  `help for list (default: false)`

`zarf destroy`

Tears down Zarf and removes its components from the environment

Synopsis

Tear down Zarf.

Deletes everything in the 'zarf' namespace within your connected k8s cluster.

If Zarf deployed your k8s cluster, this command will also tear your cluster down by searching through /opt/zarf for any scripts that start with 'zarf-clean-' and executing them. Since this is a cleanup operation, Zarf will not stop the teardown if one of the scripts produce an error.

If Zarf did not deploy your k8s cluster, this command will delete the Zarf namespace, delete secrets and labels that only Zarf cares about, and optionally uninstall components that Zarf deployed onto the cluster. Since this is a cleanup operation, Zarf will not stop the uninstalls if one of the resources produce an error while being deleted.

```
zarf destroy --confirm [flags]
```

Options

- --confirm
  `REQUIRED. Confirm the destroy action to prevent accidental deletions (default: false)`
- - h, –help
  `help for destroy (default: false)`
- --remove-components
  `Also remove any installed components outside the zarf namespace (default: false)`

`zarf init`

Prepares a k8s cluster for the deployment of Zarf packages

Synopsis

Injects a docker registry as well as other optional useful things (such as a git server and a logging stack) into a k8s cluster under the 'zarf' namespace to support future application deployments.
If you do not have a k8s cluster already configured, this command will give you the ability to install a cluster locally.

This command looks for a zarf-init package in the local directory that the command was executed from. If no package is found in the local directory and the Zarf CLI exists somewhere outside of the current directory, Zarf will failover and attempt to find a zarf-init package in the directory that the Zarf binary is located in.

```
zarf init [flags]
```

Options

- --artifact-push-token
  `[alpha] API Token for the push-user to access the artifact registry`
- --artifact-push-username
  `[alpha] Username to access to the artifact registry Zarf is configured to use. User must be able to upload package artifacts.`
- --artifact-url
  `[alpha] External artifact registry url to use for this Zarf cluster`
- --components
  `Specify which optional components to install.  E.g. --components=git-server,logging`
- --confirm
  `Confirms package deployment without prompting. ONLY use with packages you trust. Skips prompts to review SBOM, configure variables, select optional components and review potential breaking changes. (default: false)`
- --git-pull-password
  `Password for the pull-only user to access the git server`
- --git-pull-username
  `Username for pull-only access to the git server`
- --git-push-password
  `Password for the push-user to access the git server`
- --git-push-username
  `Username to access to the git server Zarf is configured to use. User must be able to create repositories via 'git push' (default: zarf-git-user)`
- --git-url
  `External git server url to use for this Zarf cluster`
- - h, –help
  `help for init (default: false)`

- --nodeport
  `Nodeport to access a registry internal to the k8s cluster. Between [30000-32767] (default: 0)`
- --registry-pull-password
  `Password for the pull-only user to access the registry`
- --registry-pull-username
  `Username for pull-only access to the registry`
- --registry-push-password
  `Password for the push-user to connect to the registry`
- --registry-push-username
  `Username to access to the registry Zarf is configured to use (default: zarf-push)`
- --registry-secret
  `Registry secret value`
- --registry-url
  `External registry url address to use for this Zarf cluster`
- --set
  `Specify deployment variables to set on the command line (KEY=value) (default: [])`
- --storage-class
  `Specify the storage class to use for the registry and git server.  E.g. --storage-class=standard`

`zarf package`

Zarf package commands for creating, deploying, and inspecting packages

Options

- - h, –help
  `help for package (default: false)`
- --oci-concurrency
  `Number of concurrent layer operations to perform when interacting with a remote package. (default: 3)`

`zarf package create`

Creates a Zarf package from a given directory or the current directory

Synopsis

Builds an archive of resources and dependencies defined by the 'zarf.yaml' in the specified directory. Private registries and repositories are accessed via credentials in your local '~/.docker/config.json', '~/.git-credentials' and '~/.netrc'.

```
zarf package create [ DIRECTORY ] [flags]
```

Options

- --confirm
  `Confirm package creation without prompting (default: false)`
- --differential
  `[beta] Build a package that only contains the differential changes from local resources and differing remote resources from the specified previously built package`
- - h, –help
  `help for create (default: false)`
- - k, –key
  `Path to private key file for signing packages`
- --key-pass
  `Password to the private key file used for signing packages`
- - m, –max-package-size
  `Specify the maximum size of the package in megabytes, packages larger than this will be split into multiple parts. Use 0 to disable splitting. (default: 0)`
- - o, –output
  `Specify the output (either a directory or an oci:// URL) for the created Zarf package`
- --output-directory
  `Specify the output (either a directory or an oci:// URL) for the created Zarf package`
- --registry-override
  `Specify a map of domains to override on package create when pulling images (e.g. --registry-override docker.io=dockerio-reg.enterprise.intranet) (default: [])`
- - s, –sbom
  `View SBOM contents after creating the package (default: false)`
- --sbom-out
  `Specify an output directory for the SBOMs from the created Zarf package`
- --set
  `Specify package variables to set on the command line (KEY=value) (default: [])`
- --skip-sbom
  `Skip generating SBOM for this package (default: false)`

`zarf package deploy`

Deploys a Zarf package from a local file or URL (runs offline)

Synopsis

Unpacks resources and dependencies from a Zarf package archive and deploys them onto the target system.
Kubernetes clusters are accessed via credentials in your current kubecontext defined in '~/.kube/config'

```
zarf package deploy [ PACKAGE ] [flags]
```

Options

- --adopt-existing-resources
  `Adopts any pre-existing K8s resources into the Helm charts managed by Zarf. ONLY use when you have existing deployments you want Zarf to takeover. (default: false)`
- --components
  `Comma-separated list of components to install.  Adding this flag will skip the init prompts for which components to install`
- --confirm
  `Confirms package deployment without prompting. ONLY use with packages you trust. Skips prompts to review SBOM, configure variables, select optional components and review potential breaking changes. (default: false)`
- - h, –help
  `help for deploy (default: false)`
- - k, –key
  `Path to public key file for validating signed packages`
- --set
  `Specify deployment variables to set on the command line (KEY=value) (default: [])`
- --sget
  `Path to public sget key file for remote packages signed via cosign`
- --shasum
  `Shasum of the package to deploy. Required if deploying a remote package and "--insecure" is not provided`

`zarf package inspect`

Displays the definition of a Zarf package (runs offline)

Synopsis

Displays the 'zarf.yaml' definition for the specified package and optionally allows SBOMs to be viewed

```
zarf package inspect [ PACKAGE ] [flags]
```

Options

- - h, –help
  `help for inspect (default: false)`
- - k, –key
  `Path to a public key file that will be used to validate a signed package`
- - s, –sbom
  `View SBOM contents while inspecting the package (default: false)`
- --sbom-out
  `Specify an output directory for the SBOMs from the inspected Zarf package`

`zarf package list`

Lists out all of the packages that have been deployed to the cluster (runs offline)

Synopsis

```
zarf package list [flags]
```

Options

- - h, –help
  `help for list (default: false)`

`zarf package publish`

Publishes a Zarf package to a remote registry

Synopsis

```
zarf package publish { PACKAGE | SKELETON DIRECTORY } REPOSITORY [flags]
```

Options

- - h, –help
  `help for publish (default: false)`
- - k, –key
  `Path to private key file for signing packages`
- --key-pass
  `Password to the private key file used for publishing packages`

`zarf package pull`

Pulls a Zarf package from a remote registry and save to the local file system

Synopsis

```
zarf package pull REFERENCE [flags]
```

Options

- - h, –help
  `help for pull (default: false)`
- - k, –key
  `Path to public key file for validating signed packages`
- - o, –output-directory
  `Specify the output directory for the pulled Zarf package`

`zarf package remove`

Removes a Zarf package that has been deployed already (runs offline)

Synopsis

```
zarf package remove { PACKAGE_NAME | PACKAGE_FILE } --confirm [flags]
```

Options

- --components
  `Comma-separated list of components to uninstall`
- --confirm
  `REQUIRED. Confirm the removal action to prevent accidental deletions (default: false)`
- - h, –help
  `help for remove (default: false)`

`zarf prepare`

Tools to help prepare assets for packaging

Options

- - h, –help
  `help for prepare (default: false)`

`zarf prepare find-images`

Evaluates components in a zarf file to identify images specified in their helm charts and manifests

Synopsis

Evaluates components in a zarf file to identify images specified in their helm charts and manifests.

Components that have repos that host helm charts can be processed by providing the --repo-chart-path.

```
zarf prepare find-images [ PACKAGE ] [flags]
```

Options

- - h, –help
  `help for find-images (default: false)`
- --kube-version
  `Override the default helm template KubeVersion when performing a package chart template`
- - p, –repo-chart-path
  `If git repos hold helm charts, often found with gitops tools, specify the chart path, e.g. "/" or "/chart"`
- --set
  `Specify package variables to set on the command line (KEY=value). Note, if using a config file, this will be set by [package.create.set]. (default: [])`

`zarf prepare generate-config`

Generates a config file for Zarf

Synopsis

Generates a Zarf config file for controlling how the Zarf CLI operates. Optionally accepts a filename to write the config to.

The extension will determine the format of the config file, e.g. env-1.yaml, env-2.json, env-3.toml etc. Accepted extensions are json, toml, yaml.

NOTE: This file must not already exist. If no filename is provided, the config will be written to the current working directory as zarf-config.toml.

```
zarf prepare generate-config [ FILENAME ] [flags]
```

Options

- - h, –help
  `help for generate-config (default: false)`

`zarf prepare patch-git`

Converts all .git URLs to the specified Zarf HOST and with the Zarf URL pattern in a given FILE. NOTE: This should only be used for manifests that are not mutated by the Zarf Agent Mutating Webhook.

Synopsis

```
zarf prepare patch-git HOST FILE [flags]
```

Options

- --git-account
  `User or organization name for the git account that the repos are created under. (default: zarf-git-user)`
- - h, –help
  `help for patch-git (default: false)`

`zarf prepare sha256sum`

Generates a SHA256SUM for the given file

Synopsis

```
zarf prepare sha256sum { FILE | URL } [flags]
```

Options

- - h, –help
  `help for sha256sum (default: false)`

`zarf tools`

Collection of additional tools to make airgap easier

Options

- - h, –help
  `help for tools (default: false)`

`zarf tools archiver`

Compresses/Decompresses generic archives, including Zarf packages

Options

- - h, –help
  `help for archiver (default: false)`

`zarf tools archiver compress`

Compresses a collection of sources based off of the destination file extension.

Synopsis

```
zarf tools archiver compress SOURCES ARCHIVE [flags]
```

Options

- - h, –help
  `help for compress (default: false)`

`zarf tools archiver decompress`

Decompresses an archive or Zarf package based off of the source file extension.

Synopsis

```
zarf tools archiver decompress ARCHIVE DESTINATION [flags]
```

Options

- --decompress-all
  `Decompress all tarballs in the archive (default: false)`
- - h, –help
  `help for decompress (default: false)`
- --unarchive-all
  `Unarchive all tarballs in the archive (default: false)`

`zarf tools clear-cache`

Clears the configured git and image cache directory

Synopsis

```
zarf tools clear-cache [flags]
```

Options

- - h, –help
  `help for clear-cache (default: false)`
- --zarf-cache
  `Specify the location of the Zarf artifact cache (images and git repositories) (default: ~/.zarf-cache)`

`zarf tools download-init`

Downloads the init package for the current Zarf version into the specified directory

Synopsis

```
zarf tools download-init [flags]
```

Options

- - h, –help
  `help for download-init (default: false)`
- - o, –output-directory
  `Specify a directory to place the init package in.`

`zarf tools gen-key`

Generates a cosign public/private keypair that can be used to sign packages

Synopsis

```
zarf tools gen-key [flags]
```

Options

- - h, –help
  `help for gen-key (default: false)`

`zarf tools gen-pki`

Generates a Certificate Authority and PKI chain of trust for the given host

Synopsis

```
zarf tools gen-pki HOST [flags]
```

Options

- - h, –help
  `help for gen-pki (default: false)`
- --sub-alt-name
  `Specify Subject Alternative Names for the certificate (default: [])`

`zarf tools get-creds`

Displays a Table of credentials for deployed components. Pass a component name to get a single credential

Synopsis

Display a Table of credentials for deployed components. Pass a component name to get a single credential. i.e. 'zarf tools get-creds registry'

```
zarf tools get-creds [flags]
```

Options

- - h, –help
  `help for get-creds (default: false)`

`zarf tools kubectl`

Kubectl command. See https://kubernetes.io/docs/reference/kubectl/overview/ for more information.

Synopsis

```
zarf tools kubectl [flags]
```

Options

- --architecture
- - h, –help
  `help for kubectl (default: false)`
- --insecure
- --log-level
- --no-log-file
- --no-progress
- --tmpdir
- --zarf-cache

`zarf tools monitor`

Launches a terminal UI to monitor the connected cluster using K9s.

Synopsis

```
zarf tools monitor [flags]
```

Options

- --architecture
- - h, –help
  `help for monitor (default: false)`
- --insecure
- --log-level
- --no-log-file
- --no-progress
- --tmpdir
- --zarf-cache

`zarf tools registry`

Tools for working with container registries using go-containertools

Options

- --allow-nondistributable-artifacts
  `Allow pushing non-distributable (foreign) layers (default: false)`
- --architecture
- - h, –help
  `help for registry (default: false)`
- --insecure
  `Allow image references to be fetched without TLS (default: false)`
- --log-level
- --no-log-file
- --no-progress
- --platform
  `Specifies the platform in the form os/arch[/variant][:osversion] (e.g. linux/amd64). (default: all)`
- --tmpdir
- - v, –verbose
  `Enable debug logs (default: false)`
- --zarf-cache

`zarf tools registry catalog`

List the repos in a registry

Synopsis

```
zarf tools registry catalog REGISTRY [flags]
```

Options

- --full-ref
  `(Optional) if true, print the full image reference (default: false)`
- - h, –help
  `help for catalog (default: false)`

`zarf tools registry copy`

Efficiently copy a remote image from src to dst while retaining the digest value

Synopsis

```
zarf tools registry copy SRC DST [flags]
```

Options

- - a, –all-tags
  `(Optional) if true, copy all tags from SRC to DST (default: false)`
- - h, –help
  `help for copy (default: false)`
- - j, –jobs
  `(Optional) The maximum number of concurrent copies, defaults to GOMAXPROCS (default: 0)`
- - n, –no-clobber
  `(Optional) if true, avoid overwriting existing tags in DST (default: false)`

`zarf tools registry login`

Log in to a registry

Synopsis

```
zarf tools registry login [OPTIONS] [SERVER] [flags]
```

Options

- - h, –help
  `help for login (default: false)`
- - p, –password
  `Password`
- --password-stdin
  `Take the password from stdin (default: false)`
- - u, –username
  `Username`

`zarf tools registry ls`

List the tags in a repo

Synopsis

```
zarf tools registry ls REPO [flags]
```

Options

- --full-ref
  `(Optional) if true, print the full image reference (default: false)`
- - h, –help
  `help for ls (default: false)`
- --omit-digest-tags
  `(Optional), if true, omit digest tags (e.g., ':sha256-...') (default: false)`

`zarf tools registry pull`

Pull remote images by reference and store their contents locally

Synopsis

```
zarf tools registry pull IMAGE TARBALL [flags]
```

Options

- --annotate-ref
  `Preserves image reference used to pull as an annotation when used with --format=oci (default: false)`
- - c, –cache_path
  `Path to cache image layers`
- --format
  `Format in which to save images ("tarball", "legacy", or "oci") (default: tarball)`
- - h, –help
  `help for pull (default: false)`

`zarf tools registry push`

Push local image contents to a remote registry

Synopsis

If the PATH is a directory, it will be read as an OCI image layout. Otherwise, PATH is assumed to be a docker-style tarball.

```
zarf tools registry push PATH IMAGE [flags]
```

Options

- - h, –help
  `help for push (default: false)`
- --image-refs
  `path to file where a list of the published image references will be written`
- --index
  `push a collection of images as a single index, currently required if PATH contains multiple images (default: false)`

`zarf tools sbom`

Generates a Software Bill of Materials (SBOM) for the given package

Synopsis

Generate a packaged-based Software Bill Of Materials (SBOM) from container images and filesystems

```
zarf tools sbom [flags]
```

Options

- --architecture
- --catalogers
  `enable one or more package catalogers (default: [])`
- - c, –config
  `application config file`
- --exclude
  `exclude paths from being scanned using a glob expression (default: [])`
- --file
  `file to write the default report output to (default is STDOUT)`
- - h, –help
  `help for sbom (default: false)`
- --insecure
- --log-level
- --name
  `set the name of the target being analyzed`
- --no-log-file
- --no-progress
- - o, –output
  `report output format, options=[syft-json cyclonedx-xml cyclonedx-json github-json spdx-tag-value spdx-json syft-table syft-text template] (default: [syft-table])`
- --platform
  `an optional platform specifier for container image sources (e.g. 'linux/arm64', 'linux/arm64/v8', 'arm64', 'linux')`
- - q, –quiet
  `suppress all logging output (default: false)`
- - s, –scope
  `selection of layers to catalog, options=[Squashed AllLayers] (default: Squashed)`
- --source-name
  `set the name of the target being analyzed`
- --source-version
  `set the name of the target being analyzed`
- - t, –template
  `specify the path to a Go template file`

- --tmpdir
- - v, –verbose
  `increase verbosity (-v = info, -vv = debug) (default: 0)`
- --zarf-cache

`zarf tools sbom attest`

Generate an SBOM as an attestation for the given [SOURCE] container image

Synopsis

Generate a packaged-based Software Bill Of Materials (SBOM) from a container image as the predicate of an in-toto attestation that will be uploaded to the image registry

```
zarf tools sbom attest --output [FORMAT] <IMAGE> [flags]
```

Options

- --catalogers
  `enable one or more package catalogers (default: [])`
- --exclude
  `exclude paths from being scanned using a glob expression (default: [])`
- --file
  `file to write the default report output to (default is STDOUT)`
- - h, –help
  `help for attest (default: false)`
- - k, –key
  `the key to use for the attestation`
- --name
  `set the name of the target being analyzed`
- - o, –output
  `report output format, options=[syft-json cyclonedx-xml cyclonedx-json github-json spdx-tag-value spdx-json syft-table syft-text template] (default: [syft-table])`
- --platform
  `an optional platform specifier for container image sources (e.g. 'linux/arm64', 'linux/arm64/v8', 'arm64', 'linux')`
- - s, –scope
  `selection of layers to catalog, options=[Squashed AllLayers] (default: Squashed)`
- --source-name
  `set the name of the target being analyzed`
- --source-version
  `set the name of the target being analyzed`
- - t, –template
  `specify the path to a Go template file`

`zarf tools sbom convert`

Convert between SBOM formats

Synopsis

[Experimental] Convert SBOM files to, and from, SPDX, CycloneDX and Syft's format. For more info about data loss between formats see https://github.com/anchore/syft#format-conversion-experimental

```
zarf tools sbom convert [SOURCE-SBOM] -o [FORMAT] [flags]
```

Options

- --catalogers
  `enable one or more package catalogers (default: [])`
- --exclude
  `exclude paths from being scanned using a glob expression (default: [])`
- --file
  `file to write the default report output to (default is STDOUT)`
- - h, –help
  `help for convert (default: false)`
- --name
  `set the name of the target being analyzed`
- - o, –output
  `report output format, options=[syft-json cyclonedx-xml cyclonedx-json github-json spdx-tag-value spdx-json syft-table syft-text template] (default: [syft-table])`
- --platform
  `an optional platform specifier for container image sources (e.g. 'linux/arm64', 'linux/arm64/v8', 'arm64', 'linux')`
- - s, –scope
  `selection of layers to catalog, options=[Squashed AllLayers] (default: Squashed)`
- --source-name
  `set the name of the target being analyzed`
- --source-version
  `set the name of the target being analyzed`
- - t, –template
  `specify the path to a Go template file`

`zarf tools sbom login`

Log in to a registry

Synopsis

```
zarf tools sbom login [OPTIONS] [SERVER] [flags]
```

Options

- - h, –help
  `help for login (default: false)`
- - p, –password
  `Password`
- --password-stdin
  `Take the password from stdin (default: false)`
- - u, –username
  `Username`

`zarf tools sbom packages`

Generate a package SBOM

Synopsis

Generate a packaged-based Software Bill Of Materials (SBOM) from container images and filesystems

```
zarf tools sbom packages [SOURCE] [flags]
```

Options

- --catalogers
  `enable one or more package catalogers (default: [])`
- --exclude
  `exclude paths from being scanned using a glob expression (default: [])`
- --file
  `file to write the default report output to (default is STDOUT)`
- - h, –help
  `help for packages (default: false)`
- --name
  `set the name of the target being analyzed`
- - o, –output
  `report output format, options=[syft-json cyclonedx-xml cyclonedx-json github-json spdx-tag-value spdx-json syft-table syft-text template] (default: [syft-table])`
- --platform
  `an optional platform specifier for container image sources (e.g. 'linux/arm64', 'linux/arm64/v8', 'arm64', 'linux')`
- - s, –scope
  `selection of layers to catalog, options=[Squashed AllLayers] (default: Squashed)`
- --source-name
  `set the name of the target being analyzed`
- --source-version
  `set the name of the target being analyzed`
- - t, –template
  `specify the path to a Go template file`

`zarf tools sbom version`

show the version

Synopsis

```
zarf tools sbom version [flags]
```

Options

- - h, –help
  `help for version (default: false)`
- - o, –output
  `format to show version information (available=[text, json]) (default: text)`

`zarf tools wait-for`

Waits for a given Kubernetes resource to be ready

Synopsis

By default Zarf will wait for all Kubernetes resources to be ready before completion of a component during a deployment.
This command can be used to wait for a Kubernetes resources to exist and be ready that may be created by a Gitops tool or a Kubernetes operator.
You can also wait for arbitrary network endpoints using REST or TCP checks.

```
zarf tools wait-for { KIND | PROTOCOL } { NAME | SELECTOR | URI } { CONDITION |
HTTP_CODE } [flags]
```

Options

- --architecture
- - h, –help
  `help for wait-for (default: false)`
- --insecure
- --log-level
- - n, –namespace
  `Specify the namespace of the resources to wait for.`
- --no-log-file
- --no-progress
- --timeout
  `Specify the timeout duration for the wait command. (default: 5m)`
- --tmpdir
- --zarf-cache

`zarf version`

Shows the version of the running Zarf binary

Synopsis

Displays the version of the Zarf release that the current binary was built from.

```
zarf version [flags]
```

Options

- - h, –help
  `help for version (default: false)`