Pulumi Infrastructure as code

CONCEPTS

Your program declares the desired resources in a real language. Pulumi diffs that against the stack's stored state and applies only the difference.

Program code per project
↓ preview
Engine diff vs state
↓ up
Stack one environment
↓ provider API
Cloud resources real infrastructure

A project is the code (Pulumi.yaml); a stack is one deployed instance of it (dev, prod) with its own config and state (Pulumi.<stack>.yaml).

Preview before up on shared stacks, and use pulumi refresh if someone changed resources by hand.

PROJECTS & DEPLOYS

`pulumi new <template>`
Create a project from a template

`pulumi new <template> --yes`
Accept all defaults

`pulumi preview`
Show what up would change

`pulumi preview --diff`
Preview as a rich diff

`pulumi up`
Preview, confirm, then deploy

`pulumi up --yes`
Deploy without the prompt

`pulumi up --refresh`
Refresh state before deploying

`pulumi up --diff`
Show changes as a rich diff

`pulumi destroy`
Delete all resources in the stack

`pulumi destroy --remove`
Also remove the empty stack

`pulumi refresh`
Update state from the real cloud

TARGETING

`pulumi up --target <urn>`
Update only these resources

`pulumi up --replace <urn>`
Force recreating a resource

`pulumi stack --show-urns`
List resources with their URNs

STACKS

`pulumi stack init <name>`
Create a stack

`pulumi stack ls`
List stacks

`pulumi stack select <name>`
Switch the current stack

`pulumi stack`
Show current stack and resources

`pulumi stack output`
Show all outputs

`pulumi stack output <name>`
Print one output value

`pulumi stack output --json`
Outputs as JSON

`pulumi stack rm <name>`
Delete an empty stack

CONFIG & SECRETS

`pulumi config`
List config values

`pulumi config set <key> <value>`
Set a config value

`pulumi config set --secret <key> <value>`
Store a value encrypted

`pulumi config get <key>`
Read one value

BACKENDS & STATE

`pulumi login`
Log in to Pulumi Cloud

`pulumi login s3://<bucket>`
Use an S3 bucket as backend

IMPORT & LOGS

`pulumi import <type> <name> <id>`
Adopt an existing resource