NAAM

kz – Opdracht uitvoeren.

SAMENVATTING

kz *OPDRACHT* [*ARGUMENT*...]
kz OPTIE

BESCHRIJVING

kz voert *OPDRACHT* uit, geïnstalleerd door pakket kz, eventueel met de opgegeven *ARGUMENT*en.

Pakket kz is een verzameling documenten en scripts voor het installeren, instellen, en beheren van bijvoorbeeld Ubuntu Linux.

Gebruik optie **-l**, **--list** voor een lijst van beschikbare *OPDRACHT*en.

OPTIES

-l, **--list** Toon de beschikbare opdrachten.

-u, **--usage**
Toon de beschikbare opties.

-h, **--help**
Toon een korte omschrijving van de beschikbare opties.

-v, **--version**
Toon de versie, de auteur, en het auteursrecht.

KZ OPDRACHTEN

De opdrachten van kz zijn verdeeld in opdrachten op hoog niveau ("porcelein", "porcelain") en opdrachten op laag niveau ("loodgieterswerk", "plumbing").

OPDRACHTEN OP HOOG NIVEAU (PORCELEIN)

Deze opdrachten zijn uit te voeren in een grafische werkomgeving met het Installatiemenu of door te zoeken naar de opdracht, en in het Terminalvenster.

kz-backup(1)
Back-up maken.

kz-install(1)
Apps installeren.

kz-menu(1)
Installatiemenu.

kz-restore(1)
Back-up terugzetten.

kz-setup(1)
Apps instellen.

kz-wifi(1)
Wifi-informatie tonen.

OPDRACHTEN OP LAAG NIVEAU (LOODGIETERSWERK)

Deze opdrachten zijn alleen uit te voeren in het Terminalvenster en zijn ondersteunend voor het beheer en gebruik van kz en Linux.

kz-build(1)
Pakket kz bouwen.

kz-ckname(1)
Bestandsnamen controleren.

kz-deploy(1)
Pakket kz distribueren.

kz-getdeb(1)
Pakket kz installeren.

kz-getdev(1)
Ontwikkelomgeving bouwen.

kz-gset(1)
GUI wijzigen.

kz-ivp(1)
Scripts controleren.

kz-update(1)
Systeem bijwerken.

STOPSTATUS

Normaal wordt afsluitwaarde 0 teruggegeven; als er een fout optreedt, is de afsluitwaarde ongelijk 0.

NOTITIES

Pakket **kz** geeft met de documenten en scripts invulling aan en ondersteuning voor:
1. IaC
2. CI/CD
3. Day 0/Day 1/Day 2 Operations
Begrippen uit de moderne wereld van software-ontwikkeling, -distributie, en beheer.

1. IaC (Infrastructure as Code, Infrastructuur als Code of programmeerbare infrastructuur)
Het definiëren en beheren van infrastructuur door middel van code.
De opdrachten die standaard meegeleverd worden bij pakket kz zijn idempotent; herhaaldelijk utvoeren leidt steeds tot hetzelfde resultaat.
Herhaalbaarheid (idempotentie) is één van de principes van IaC.
De documenten en scripts worden vastgelegd en onderhouden in versiebeheer (GitHub), een ander principe van IaC.
Voorbeelden zijn **kz-install-ubuntu-desktop.sh** (installatiebestand gebruikt door **kz install) en kz-setup-ubuntu-desktop.sh** (instelbestand gebruikt door **kz setup**).

2. CI/CD (Continuous Integration and Delivery, Continue integratie en levering)
CI/CD is de gecombineerde praktijk van continuous integration (CI) en (vaker) continuous delivery of (minder vaak) continuous deployment (CD).
CI (Continuous Integration, Continue integratie) is het meerdere keren per dag de werkkopieën van alle ontwikkelaars samenvoegen tot een gedeelde hoofdlijn.
CD (Continuous Delivery, Continue levering) is software produceren in korte cycli, ervoor zorgen dat de software op elk moment betrouwbaar kan worden vrijgegeven en, bij het vrijgeven van de software, zonder dit handmatig te doen.
CD (Continuous Deployment, Continue doorzetten) is een aanpak waarbij softwarefunctionaliteiten vaak en via geautomatiseerde implementaties worden geleverd.
Een **CI/CD Pipeline** is een reeks stappen die moet worden uitgevoerd om geautomatiseerd een nieuwe softwareversie te leveren.
Voorbeelden zijn **kz build** en **kz deploy**.

CI/CD Pipeline flow
Applicatiecode ----> Source scm ----> Build ----> Test ----> Deployment
vscode GitHub kz-build CLI/TUI/GUI kz-deploy
+---------- DEV ----------+ +------- TEST --------+ +- PROD -+

3. Day 0/Day 1/Day 2 Operations
In IT verwijzen de termen Day 0/Day 1/Day 2 Operations naar verschillende fasen van software-

ontwikkeling, -distributie, en beheer.

3.1 Day 0 Operations (Dag 0 Activiteiten)
Dit is de ontwerpfase (design), waarin projectvereisten worden gespecificeerd en de architectuur van de oplossing wordt bepaald.
Hierbij gebruik ik Visual Studio Code en GitHub, en programmeer ik in Bash en Python.

3.2 Day 1 Operations (Dag 1 Activiteiten)
Omvat het ontwikkelen en implementeren (development and deploy) van software die is ontworpen in de dag 0-fase.
Hierbij wordt onder andere gebruik gemaakt van **IaC** en **CI/CD Pipelines**, en het uitvoeren van Ansible Playbooks.
Voorbeelden zijn Checklist installatie, en **kz getdeb** en **kz menu**.

3.3 Day 2 Operations (Dag 2 Activiteiten)
Hier gaat de meeste aandacht uit naar het onderhouden, bewaken en optimaliseren (maintaining, monitoring, and optimizing) van het systeem.
Voorbeelden zijn **kz backup** en **kz update**.

VOORBEELDEN

kz update
Werk systeem bij.

kz install google-chrome
Installeer Google Chrome.

kz setup --cat google-chrome
Toon instel-opdrachten voor Google Chrome.

AUTEUR

Geschreven in 2021 door Karel Zimmer <info@karelzimmer.nl>, Creative Commons Publiek Domein Verklaring <https://creativecommons.org/publicdomain/zero/1.0>.

ZIE OOK

https://karelzimmer.nl

KZ

Onderdeel van het **kz**(1) pakket, genoemd naar de maker Karel Zimmer.

BESCHIKBAARHEID

Opdracht **kz** is onderdeel van het pakket **kz** en is beschikbaar op de website van Karel Zimmer <https://karelzimmer.nl/html/nl/linux.html#scripts>.