

# Poglavje 1 Okolje OpenStack

V tem poglavju si bomo podrobneje pogledali platformo OpenStack, ki jo bomo v naslednjem poglavju tudi namestili. Zato je potrebno obravnavanje delovanja platforme in funkcionalnosti storitev.

## 1.1 Predstavitev OpenStacka

OpenStack je odprtokodna platforma, ki omogoča postavitev oblaka. Platforma je tudi projekt, ki nastaja na podlagi potreb velikih korporacij, ki ga dodelujejo svojim potrebam. Po storitvenem modelu oblaka infrastrukturo ponuja kot storitev, po načinu postavitve je lahko javen ali zaseben. Zadnja inačica platforme je Juno. V okviru tega diplomskega dela je predstavljena inačica Icehouse. Na vsakih šest mesecev izide nova verzija platforme. Smernice razvoja platforme potekajo v večji modularnosti posameznih komponent, ki se bodo lahko samostojno uporabljale v novejših storitvenih modelih računalništva v oblaku. OpenStack je večinoma implementiran v programskem jeziku Python. Zaradi številnih prednosti smo se odločili uporabiti omenjeno platformo:

- omogoča različni načine postavljanja in vzdrževanje oblaka;
- zanesljivo delovanje in varnost podatkov;
- številne možnosti razširljivosti in prilagodljivosti;
- velik nabor hipernadzornikov in strojne opreme;
- uporaba spletne konzole namesto ukazne vrstice;
- velika odprtokodna skupnost in podpora s strani velikih računalniških podjetij;
- določene komponente je mogoče uporabiti kot samostojne komponente v integraciji z ostalimi rešitvami.

Tudi številna podjetja so prepoznala uporabo in se vključila v razvoj med več kot 350 podjetij so najbolj znani: Intel, HP, IBM in Cisco.

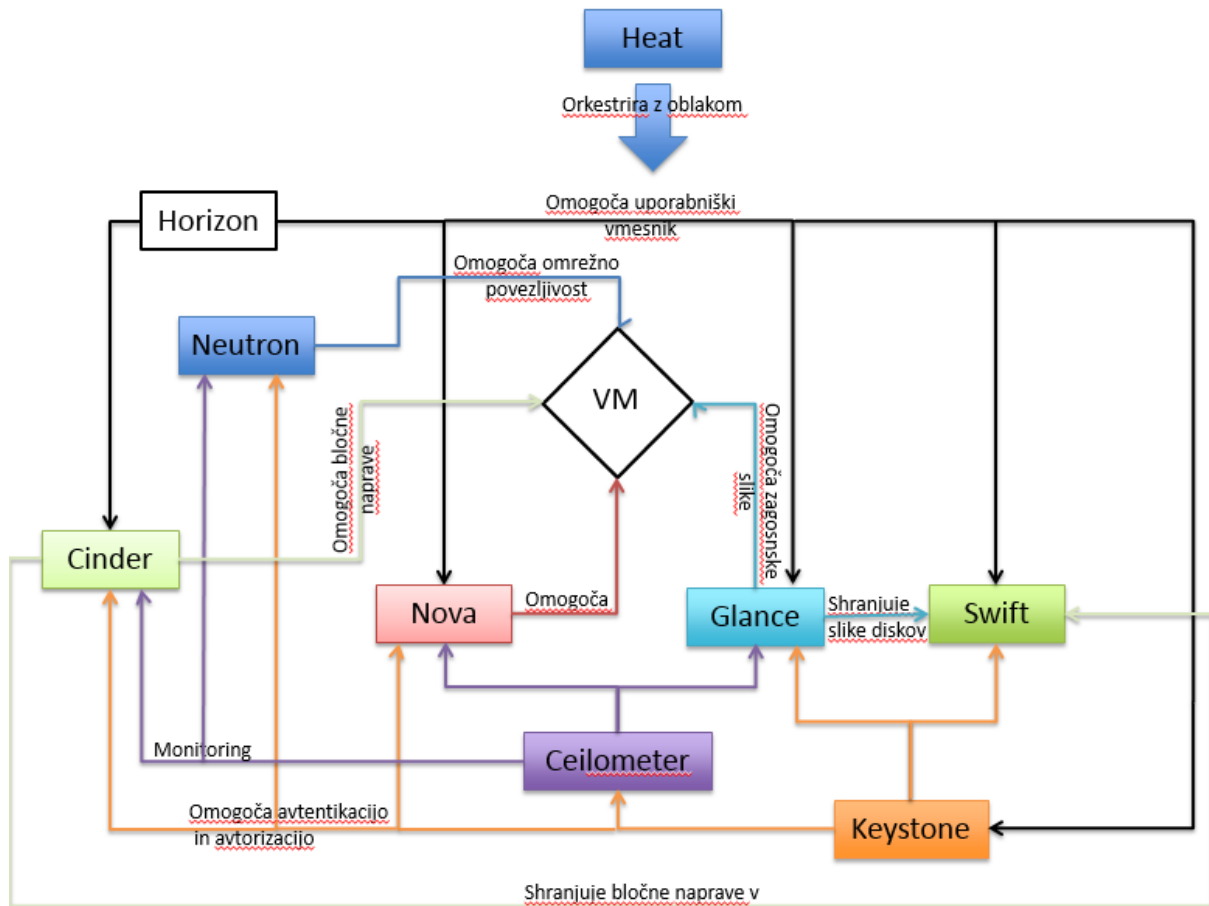
Verzija Icehouse tvori deset komponent oziroma modulov. Za osnovno postavitve oblaka je potrebno namesti prve tri komponente OpenStacka. Vsaka komponenta predstavlja storitev, ki jo platforma uporablja za delovanje:

- *Nova* - procesna (računska) storitev (*ang. compute service*);
- *Keystone* - identifikacijska storitev (*ang. identity service*);
- *Glance* - storitev za shranjevanje slik diskov navideznih strojev (*ang. image service*);
- *Neutron* - omrežna storitev (*ang. network service*);
- *Horizon* - spletni vmesnik oziroma nadzorna plošča platforme (*ang. dashboard*);
- *Cinder* - storitev za trajno shranjevanje podatkov v obliki blokovnih naprav (*ang. block storage service*);
- *Swift* - storitev, ki omogoča trajno shranjevanje podatkov v obliki objektov (*ang. object storage service*);
- *Heat* - orkestracijska storitev (*ang. orchestration service*);
- *Ceilometer* - modul za spremljanje in merjenje porabe virov (*ang. telemetry module*);
- *in Trove* - ponuja povečljivo in zanesljivo storitev podatkovne baze (*ang. cloud database as a service*).

## 1.2 Konceptualna arhitektura OpenStacka

Konceptualna arhitektura, ki je podana na Sliki 1, podaja osnovne relacije med komponentami platforme. Uporabniki in skrbniki sistemov lahko preko spletnega vmesnika Horizon manipulirajo s storitvami in z viri platforme, lahko pa uporabijo vmesnik CLI. Keystone vsem uporabnikom in storitvam omogoča avtentikacijo ter avtorizacijo, ki jo potrebujejo za uporabo storitev oziroma za medsebojno komunikacijo. Nova izvaja osnovne operacije nad navideznimi stroji in jedrni del platforme. Glance shranjuje zagonke slike operacijskih sistemov, ki se uporabijo pri zagonu navideznega stroja. Tudi takrat, ko naredimo posnetek navideznega stroja (*ang. snapshot*), ga ta storitev shrani. Pri tem se podatki stroja shranijo znotraj slike. Cinder omogoča trajen pomnilnik navideznim strojem v obliki blokovne naprave (*ang. volume*), ki jo je potrebno pripeti na navidezni stroj in nastaviti datotečni sistem. V primeru uničenja stroja ali izpada se zažene nov navidezni stroj, h kateremu se pripne blokovna naprava. Tako so vsi podatki, ki se nahajajo na omenjeni napravi, tudi po uničenju ali izpadu navideznega stroja na voljo. Swift omogoča tako shranjevanje slik, ki jih omogoča Glance, kot nestrukturnih podatkov

v obliki objektov. Neutron omogoča navidezno omrežje, v katerega se povežejo navidezni stroji; tako omogoča njihovo uporabo preko spleta. Ceilometer izvaja monitoring in meritve storitve za njihovo zaračunavanje uporabnikom. Heat preko v naprej pripravljenih šablon omogoča navidezno infrastrukturo. Druga možnost upravljanja platforme je preko klientov ukazne vrstice posameznih komponent, ki omogoča pisanje ukazov ali pa poganjanje skript.



Slika 1: Konceptualna arhitektura OpenStacka.

### 1.3 Logična arhitektura platforme OpenStacka

Logična arhitektura je bolj zapletena in podaja natančen pregled API-jev posameznih komponent, preko katerega se je mogoče povezati in uporabljati storitve ostalih komponent platforme. Tako so komponente sinhronizirane in oblak deluje kot celota. Komponente platforme delujejo po enem izmed spodaj opisanih načinov, in sicer:

- prikriti proces (*ang. daemon*) teče v ozadju storitve in sprejema zahteve ter se nanje odziva;

- skripte (*ang. script*) se uporabljajo za namestitvev okolja platforme in za zaganjanje testov za preverjanje delovanja komponent;
- CLI (*ang. command-line interface*) uporabnikom in skrbnikom sistema omogoča pošiljanje zahteve preko vmesnika API do storitev, ki jih uporablja in ponuja platforma.

V Logična arhitektura je podana shema logične arhitekture platforme, na kateri si je mogoče ogledati relacije med API-ji in komponentami. Zaradi visoke prilagodljivosti platforme je lahko končna logična arhitektura precej drugačna in v veliki meri odvisna od v naprej določene uporabe oblaka oziroma spreminjajočih se potreb podjetja. Podroben opis komponent in API-jev je podan v naslednjem poglavju.