

Rīgas Mežu izsoļu sistēmas servera administrācijas instrukcija

v. 1.0

Datums: 28.05.2024

Sagatavoja: Kalvis Buls (kalvis.buls@ecomtech.lv)

SIA Ecommerce Technologies

Saturs

Saturs	2
Sistēmas pārvaldīšana	3
Sistēmas sagatavošana.....	3
Projekta struktūra.....	3
Cron.....	3
Fona procesi (queue).....	3
Versionēšana	4
Piegādes process	5
Piegādes izveidošana.....	5
Piegādes uzstādīšana.....	5
Vecākas piegādes atkārtota uzstādīšana.....	5
Aplikācijas konfigurācija	6
Konfigurācijas pārvaldīšana.....	6
Pieejamie konfigurācijas lauki.....	7

Sistēmas pārvaldīšana

Sistēmas sagatavošana

Visa nepieciešamā sistēmas sagatavošana tiek veikta automātiski, izmantojot Ansible (<https://www.ansible.com/>).

Projekta struktūra

Pieņemot, ka projekts ir uzstādīts mapē `/home/auctions/app`, tiks izmantota sekojoša struktūra:

- `/home/auctions/app/shared` - kopīgie faili, piemēram, statiskā konfigurācija, augšupielādētie faili, sistēmas logu faili u.c.
- `/home/auctions/app/current` - pašreizējā piegāde (*symbolic link*).
- `/home/auctions/app/releases` - visas piegādes.

Pašreizējās versijas *symbolic link* tiks atjaunots piegādes uzstādīšanas laikā.

Cron

Cron (<https://en.wikipedia.org/wiki/Cron>) ir atbildīgs par automātisku procesu izpildīšanu, ja tiem ir fiksēts izpildes laiks, kas nemainās. Cron konfigurācijas saturs (*crontab*) tiek ģenerēts automātiski, izmantojot Ansible, tāpēc ikdienā tas nebūs jāmaina.

Esošo konfigurāciju var apskatīt, izpildot sekojošās konsoles komandas:

- `crontab -l` - sistēmas procesi
- `crontab -u auctions -l` - aplikācijas procesi

Fona procesi (queue)

Fona procesi tiek izmantoti automātiskai darbību izpildei, ja laiks nav fiksēts (piem., izsoles uzsākšana konfigurācijā norādītajā laikā), kā arī jebkurām asinhronām darbībām, kur nav jāgaida izpildes rezultāts (piem., e-pastu izsūtīšanai).

Fona procesu nepārtrauktas darbības nodrošināšanai tiek izmantots Supervisor (<http://supervisord.org>).

Versionēšana

Aplikācijas versionēšana notiek, izmantojot *Semantic Versioning 2.0.0* (<https://semver.org/>), kur katras versijas numurs sastāv no trīs daļām (MAJOR.MINOR.PATCH) un tiek noteikts pēc veikto izmaiņu veida:

- MAJOR - izmaiņas, kas nav saderīgas ar iepriekšējo MAJOR versiju (piem., liela mēroga izmaiņas izsoļu organizēšanas procesā).
- MINOR - jauna funkcionalitāte, kas ir saderīga ar iepriekšējo MAJOR versiju (piem., plānotās regulārās piegādes).
- PATCH - MAJOR vai MINOR versiju kļūdu labojumi (piem., piegādes testēšanas laikā atrasto kļūdu novēršana).

Papildus versijas numuram, versija var saturēt sekojošas daļas:

- Projekta prefikss - nav īpašas nozīmes un vienmēr būs statisks viena projekta līmenī (piem., *RMA_*).
- Versijas veida modifikatori - paredzēti testa vidēm versijām, kas vēl netiek uzskatītas par stabilām (piem., *alpha, beta, rc, dev* u.c.).

Palielinot katru versijas segmentu (piem., MAJOR), turpmākie segmenti vienmēr tiks atiestatīti uz 0. Piemēram, ja pašreizējā versija ir 1.15.2, tad atkarībā no veiktajām izmaiņām nākamās versijas būtu attiecīgi:

- MAJOR - 2.0.0
- MINOR - 1.16.0
- PATCH - 1.15.3

Katra nākamā versija ir balstīta uz iepriekšējo, tātad sevī ietver visu iepriekšējo versiju saturu, tādēļ, veicot piegādes uzstādīšanu, nav nepieciešams atkārtoti uzstādīt iepriekšējās versijas.

Piegādes process

Piegādes izveidošana

- Projekta Git repozitorijā (<https://git.ecomtech.lv/rigas-mezi/auctions/>) *main* zarā tiek iekļauti visi piegādē paredzētie JIRA pieteikumi, apstiprinot attiecīgos GitLab MR (*merge request*).
- Uz *main* zara bāzes tiek izveidots jauns Git piegādes tags, kas sakrīt ar JIRA piegādes versiju, piem., RMA_1.16.0.
- GitLab CI/CD automātiski tiek izveidots arhīvs ar jaunās versijas saturu (iekļauj projekta kodu, kā arī visas nepieciešamās trešās puses bibliotēkas).

Piegādes uzstādīšana

- Gan testa vidē, gan produkcijā piegādes tiek manuāli uzstādītas no GitLab CI/CD pēc tam, kad ir izveidots attiecīgās piegādes arhīvs.
- Piegādes uzstādīšanas laikā tiek veiktas nepieciešamās darbības, lai aplikācija varētu darboties uzreiz pēc uzstādīšanas, ieskaitot:
 - Datubāzes shēmas migrācijas.
 - Sākotnējās administrācijas paneļa konfigurācijas izveidošana jaunajiem laukiem.
 - Jauno tulkojumu importēšana.
- Pēc piegādes uzstādīšanas tajā iekļautie JIRA pieteikumi tiek atzīmēti kā piegādāti.
- Ecommerce Technologies atbildīgā persona nosūta e-pastu klientam, kurā norādīta uzstādītā versija, piegādē iekļautie pieteikumi, kā arī papildus informācija, ja tāda ir nepieciešama (piem., kurās vidēs piegāde ir bijusi uzstādīta).

Uzstādīšanas process ir identisks gan testa, gan produkcijas vidēm. Visi izveidotie piegāžu arhīvi tiks saglabāti sešus mēnešus kopš to izveidošanas brīža, tātad piegādi var uzstādīt jebkurā laikā, kamēr vien arhīvs ir pieejams.

Vecākas piegādes atkārtota uzstādīšana

Ja tas ir nepieciešams, ir iespējams atkārtoti uzstādīt arī vecākas piegādes, taču tikai gadījumos, ja starp vecāko un pašlaik uzstādīto piegādi nav bijušas neatgriezeniskas datubāzes shēmas izmaiņas, kā arī joprojām ir pieejams izveidotais arhīvs ar piegādes saturu.

Aplikācijas konfigurācija

Konfigurācijas pārvaldīšana

Statiskā aplikācijas konfigurācija tiek pārvaldīta, izmantojot t.s. *environment* mainīgos, kas ir konfigurējami projekta kopīgo failu mapē esošajā `.env` failā, piemēram, `/home/auctions/app/shared/.env`.

Lauki, kas ir failā, taču nav uzskaitīti zemāk esošajā tabulā, nav paredzēti mainīšanai. Daļai lauku norādītās vērtības maiņa var negatīvi ietekmēt vai pat apturēt aplikācijas darbību, savukārt citi ir paredzēti tikai sākotnējās administrācijas paneļa konfigurācijas izveidošanai, uzstādot piegādi.

Lai veiktās izmaiņas stātos spēkā, papildus nepieciešams pārlādēt aplikācijas konfigurāciju, kā arī pārstartēt fona procesus, izpildot sekojošās konsoles komandas:

- `php artisan config:clear` - jāizpilda no pašreizējās piegādes mapes
- `supervisorctl restart all` - var izpildīt no jebkuras mapes

Pieejamie konfigurācijas lauki

Nosaukums	Apraksts	Noklusētā vērtība
APP_NAME	Aplikācijas nosaukums, tiek izmantots pārlūka cilnes nosaukumam	Laravel
APP_KEY	Aplikācijas šifrēšanas atslēga	
ACTIVITY_LOGS_DELETE_AFTER_DAYS	Laiks dienās, cik ilgi datubāzē jāglabā administrācijas paneļa audita ieraksti	730 (2 gadi)
AUTHORIZATION_LOGS_DELETE_AFTER_DAYS	Laiks dienās, cik ilgi datubāzē jāglabā lietotāju autorizācijas logu ieraksti	730 (2 gadi)
DB_PASSWORD	Projekta datubāzes parole (skatīt failu <i>.my.cnf</i> projekta direktorijā)	Ģenerēta automātiski
REDIS_PASSWORD	Redis parole	
SESSION_LIFETIME	Sesijas ilgums minūtēs	
MAIL_HOST	SMTP hosts	smtp.mailgun.org
MAIL_PORT	SMTP ports	587
MAIL_USERNAME	SMTP lietotājvārds	
MAIL_PASSWORD	SMTP parole	
MAIL_ENCRYPTION	SMTP izmantotā šifrēšana	tls
MAIL_FROM_ADDRESS	E-pasta adrese izejošajiem e-pastiem	
MAIL_FROM_NAME	E-pasta autora vārds izejošajiem e-pastiem	
ADMIN_PASSWORD_RESET_EXPIRE_IN_MINUTES	Laiks minūtēs, cik ilgi ir derīgs administrācijas paneļa lietotāju paroles atiestatīšanas e-pasts	4320 (3 dienas)
ADMIN_INVITATION_EXPIRE_IN_MINUTES	Laiks minūtēs, cik ilgi ir derīgs jauna administrācijas paneļa lietotāja ielūgums	43200 (30 dienas)