Jenkins on AWS

Овој труд е архивиран

*Maj 2017*

За најнова техничка содржина, видете

https://aws.amazon.com/whitepapers/?

**Известувања**

Овој документ е даден само за информативни цели. Ги претставува актуелните понуди и практики на AWS за производот од датумот на издавање на овој документ, кои се предмет на промена без претходна најава. Клиентите се одговорни да направат сопствена независна проценка на информациите во овој документ и каква било употреба на производите или услугите на AWS, од кои секоја е обезбедена „онаква каква што е“ без гаранција од било кој вид, без разлика дали е изразена или имплицитна. Овој документ не создава гаранции, застапувања, услови за договорни обврски или гаранции од AWS, неговите филијали, добавувачи или даватели на лиценци. Одговорностите и обврските на AWS кон своите клиенти се контролираат со договори за AWS, и овој документ не е дел, ниту пак менува, каков било договор помеѓу AWS и неговите клиенти.

**Содржини**

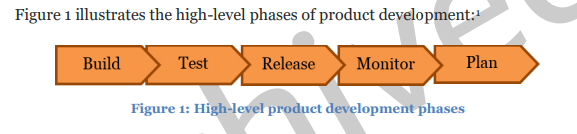
**Апстракт**

Иако постојат многу начини да се распореди серверот за автоматизација со отворен код Jenkins на веб-услугите на Амазон (AWS), оваа бела хартија се фокусира на два специфични пристапи. Прво, традиционалното распоредување на врвот на Amazon Elastic Compute Cloud (Amazon EC2). Второ, контејнерско распоредување што ја користи услугата за контејнери на Amazon EC2 (Amazon ECS). Овие пристапи им овозможуваат на клиентите да ги искористат предностите на континуираната способност за интеграција/ континуирана испорака (CI/ CD) на Jenkins. Користејќи обемен систем за додатоци, Jenkins нуди опции за интегрирање со многу AWS услуги и може да се преобликува за да одговара на повеќето случаи (на пример, традиционални развојни цевководи, развој на мобилни телефони, безбедносни барања итн.).

**Вовед**

Зошто CI/CD?

За да го разбереме моделот за континуирана интеграција/континуирана испорака (CI/CD) што го користи Jenkins, да почнеме со разбирање на неговите основни двигатели. Од раните 2000-ти, појавата на брза итеративна методологија, како што е агилната, го смени размислувањето за развој на софтвер. Во оваа нова парадигма, тимовите за производи ја туркаат својата работа до клиентите што е можно побрзо за да можат да соберат повратни информации и да се подобрат по претходното повторување на нивните производи. Концептите како што се минимален остварлив производ (MVP), кандидат за ослободување, брзина, итн. сите се изведени од овие нови пристапи. Спротивно на тоа, тимовите за производи кои користат постари парадигми како што е развојот на водопадот можеби нема да ги слушнат од клиентите со месеци и, доста често, дури додека производот не се комерцијализира.



Редоследот и должината на овие фази варираат во зависност од тоа кои модели за развој се користат (на пример, водопад, модел v, скрум, итн.).

Континуираната интеграција (CI) е практика за развој на софтвер во која програмерите редовно ги спојуваат своите промени во кодот во централно складиште, по што се извршуваат автоматски изработки и тестови. Континуираната интеграција најчесто се однесува на фазата на изградба или интеграција на процесот на ослободување на софтверот и вклучува и компонента за автоматизација (на пример, CI или услуга за изградба) и културна компонента (на пример, учење за често интегрирање). Клучните цели на континуираната интеграција се побрзо пронаоѓање и решавање на грешки, подобрување на квалитетот на софтверот и намалување на времето потребно за валидација и објавување нови ажурирања на софтверот.

Основните предизвици на континуираната интеграција вклучуваат одржување единствено складиште за изворен код, автоматско градење (и брзо градење) и автоматско тестирањето. Дополнителни предизвици вклучуваат тестирање на клон на производната средина, обезбедување видливост на процесот за тимот и овозможување на програмерите лесно да ја добијат најновата верзија.

Целта на оваа бела хартија е да ви покаже како користењето на Jenkins на AWS е стратегија погодна за решавање на овие предизвици за CI.

**Континуирана испорака и распоредување**

Континуираната испорака (CD) е практика за развој на софтвер каде промените на кодот автоматски се градат, тестираат и подготвуваат за издавање во продукција. Се проширува при континуирана интеграција со распоредување на сите промени на кодот во средина за тестирање, производна средина или и двете по завршувањето на фазата на изградба. Кога континуираната испорака е правилно имплементирана, програмерите секогаш имаат артефакт за изградба подготвен за распоредување што поминал низ стандардизиран тест процес.

Со континуирано распоредување, ревизиите се распоредуваат во производна средина автоматски без експлицитно одобрување од програмерот, што го прави целиот процес на издавање софтвер автоматизиран. Ова, од своја страна, овозможува производот да биде пред своите клиенти рано и повратните информации да почнат да се враќаат во развојните тимови.

**Зошто да користите Jenkins?**

Jenkins е многу популарен производ меѓу клиентите на AWS кои сакаат да ги автоматизираат своите CI/CD цевководи.2

* Ги остварува сите фази опишани во претходниот дел.
* Многу добро се интегрира на јазици, платформи и операциони системи.
* Тоа е софтвер со отворен код.

Jenkin работи добро на AWS и со AWS бидејќи е достапен на пазарот AWS; 3 тоа е широко документирано; и тоа е многу добро интегрирано. Дополнително, додатоците на Jenkins се достапни за голем број AWS услуги. Остатокот од оваа бела хартија дискутира за некои од тие додатоци и за она што им дозволуваат на нашите клиенти да го постигнат.

**Распоредување на Jenkins на AWS**

Во овој дел разговараме за два пристапа за распоредување на Jenkins на AWS. Прво, можете да го користите традиционалното распоредување на врвот на Amazon Elastic Compute Cloud (Amazon EC2). Второ, можете да го користите контејнерското распоредување што ја користи услугата за контејнери на Amazon EC2 (Amazon ECS). И двата пристапи се производствено подготвени за опкружување на претпријатието. Покрај тоа, двата пристапа ја поставуваат околината на Jenkins во Амазон Виртуелен приватен облак (Амазон VPC).

**Amazon VPC**

Amazon VPC ви овозможува да обезбедите логично изолиран дел од AWS Cloud каде што можете да стартувате AWS ресурси во виртуелна мрежа што ќе ја дефинирате. Имате целосна контрола врз вашата виртуелна околина за вмрежување, вклучувајќи избор на сопствен опсег на IP адреси, создавање подмрежи и конфигурација на табели за маршрути и мрежни порти.

Ви препорачуваме да ја стартувате инсталацијата на Jenkins во рамките на VPC. Лансирањето во VPC не само што ви овозможува да ги држите ресурсите на Jenkins одвоени од другите ресурси што може да ги користите, туку исто така обезбедува можност да имате контрола над IP -адресите на статички примери, што ќе биде важно додека додавате работнички јазли во вашата конфигурација (повеќе за тоа подоцна).

**Традиционално распоредување**

Традиционалното распоредување значи распоредување на Jenkins на врвот на Амазон EC2. Подоцна во оваа бела хартија истражуваме распоредување со контејнери.

**Преглед на архитектурата на Jenkins**

Архитектурата на Jenkins е прилично јасна. Надвор од кутијата, тој е распореден и како сервер и како агент за изградба што работи на ист домаќин. Можете да изберете да го распоредите enенкинс како сервер или агент за изградба, што овозможува раздвојување на оркестрацијата и изградба на извршување. Ова, пак, овозможува поголема флексибилност на дизајнот на архитектурата.

**Стратегии за распоредување на работни јазли на Jenkins**

Стандардно, Jenkins серверот ќе ги обработува сите HTTP барања, како и градбите за секој проект. Како што расте бројот на корисници, или се зголемува количината или сложеноста на работните места, главниот сервер може да доживее деградирани перформанси поради оданочување на ресурсите како процесорот и меморијата, или поради бројот на градби што се извршуваат на главниот сервер.

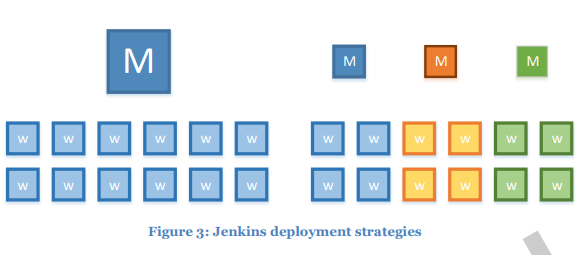
Ова е кога агентите за изградба (или јазли на работници) можат да имаат корист од инсталацијата на Jenkins со ослободување на ресурсите на главниот јазол и обезбедување прилагодени средини во кои ќе се тестираат градбите. Работнички јазол содржи агент кој комуницира со главниот сервер и работи со лесна градба на Jenkins што му овозможува да прима и работи без работа.



**Стратегии за Jenkins Master Deployments**

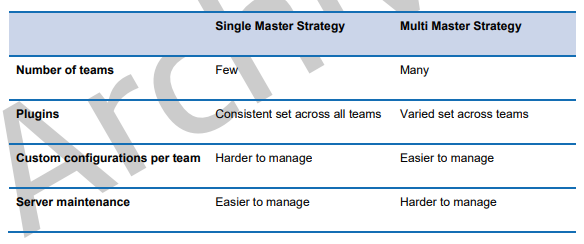
Инсталациите на Jenkins генерално спаѓаат во едно од двете сценарија:

1. Еден, голем главен сервер со повеќе работни јазли поврзани со него.
2. Повеќе помали главни сервери со повеќе работни јазли поврзани со секој од нив



Во двата случаи, еден или повеќе работни јазли се присутни. Во поголемите системи, ова е важна практика – не гради на master. Изборот помеѓу еден master или повеќе masters зависи од неколку фактори, но обично гледаме дека клиентите усвојуваат повеќе masters. На пример, Netflix работи повеќе од 25 masters на AWS.

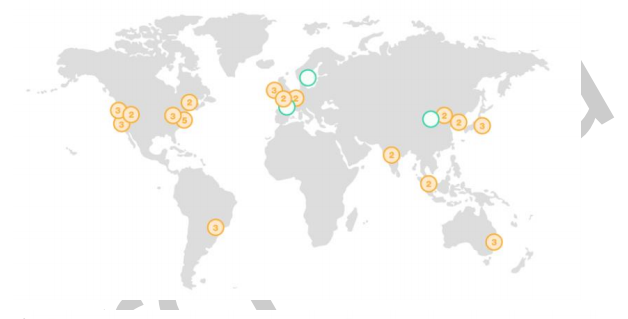
Следната табела дава некои критериуми што можете да ги користите кога избирате која стратегија најдобро одговара на вашите потреби:

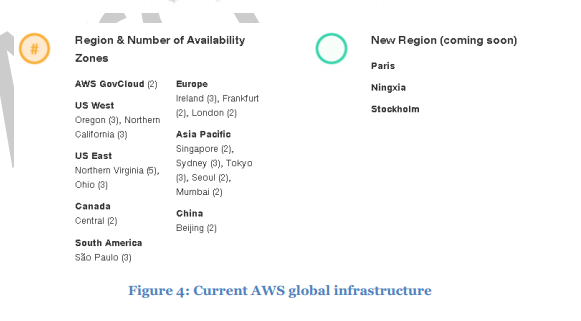


Иако постојат многу успешни стратегии за распоредување на повеќе masters на Jenkins, овој труд се фокусира на распоредување на еден master на Jenkins на Amazon EC2, но со архитектирање на тоа да биде само-заздравувачки и високо достапен. Техниките опишани овде за распоредување на еден master на Jenkins може да се применат за да се создаде средина со повеќе masters.

**Архитектура за High Availability**

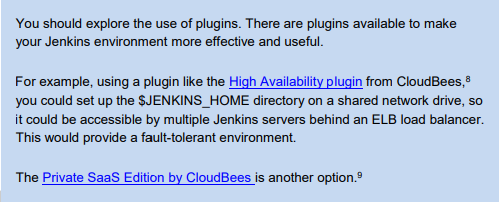
Инфраструктурата AWS (Слика 4) е изградена околу региони и зони на достапност (AZs). Регион е физичка област на светот каде AWS има повеќе Достапни зони. Достапна зона се состои од еден или повеќе дискретни центри за податоци, секој со вишок енергија, вмрежување и поврзување, сместени во посебни капацитети. Овие Достапни зони ви нудат можност да работите со производствени апликации и бази на податоци што се повеќе достапни, толерантни за грешки и поскалабилни отколку што би било можно од еден центар за податоци.





Во AWS Cloud-от, веб-пристапна апликација како Jenkins е типично дизајнирана за голема достапност и толеранција на грешки со ширење на примери низ повеќе зони на достапност и нивно представување со балансирач на оптоварување на еластично оптоварување (ELB). Еластично балансирање на оптоварување автоматски го дистрибуира влезниот сообраќај на апликации низ повеќе инстанци на Amazon EC2 во Cloud-от. Тоа ви овозможува да постигнете поголеми нивоа на толеранција на грешки во вашите апликации и беспрекорно го обезбедува потребниот износ на капацитет за балансирање на оптоварување потребен за дистрибуција на сообраќај на апликации.

Поради фактот дека Jenkins складира главна конфигурација на јазол во $JENKINS\_HOME директориум - наместо, да речеме, база на податоци - станува проблематично да се одржува вишок Jenkins сервер во посебна Достапна зона во единствено главно поставување без помош на приклучоци. Задачите како конфигурирање на Jenkins, одржување додатоци и управување со корисници ќе треба да се повторат низ секоја реплицирана инстанца во секоја Достапна зона.



Ако вашите деловни барања бараат околина на Jenkins, толерантна за грешки, претпочитаното поставување може да биде сценарио во кое повеќе masters со свои работници се сместени во посебна зона на достапност.

Бидејќи фокусот на оваа бела хартија е на главното сценарио, треба да размислите за создавање на Amazon CloudWatch аларм што го следи вашиот пример на Jenkins. CloudWatch автоматски го обновува примерот ако се оштети поради основниот хардверски дефект или проблем што бара вклучување на AWS за поправка. Овој метод ви дава можност брзо да закрепнете во случај на неуспех без да имате корист од водење на enенкинс низ повеќе Достапни зони.

Подоцна во оваа бела хартија, предлагаме раздвојувањето на складиштето за конфигурација од пресметковниот јазол Jenkins овозможува зголемена достапност.

Овие опции не бараат дополнителни додатоци на Jenkins. Наместо тоа, тие се потпираат на технологијата AWS.

**Размислувања за ресурси за правилна големина на вашиот Jenkins Master**

Како и со секое распоредување на AWS, големината на вашиот пример во однос на процесорот, меморијата и складиштето има влијание врз перформансите и профилите на трошоци. Затоа, многу е важно да се направат вистинските избори.

**Операционен систем**

Зборуваме за распоредување на Jenkins користејќи Amazon Linux, но лесно е да се прилагодат други вкусови на Linux или околини со Windows користејќи го инсталаторот на Windows за Jenkins.

**CPU и вмрежувањето**

Распоредот на Jenkins изграден за обем и голема достапност варира врз основа на бројот на работни јазли што се поврзуваат со главен јазол. Главен јазол Jenkins отвора повеќе нишки по конекција - две за секоја SSH и Java Web Start (JWS) конекција и три за секоја HTTP конекција.

Jenkins служи содржина на табла преку HTTP. Затоа, ако очекувате голем број истовремени корисници, треба да очекувате дополнителни трошоци за CPU за изработка на оваа содржина.

**Складирање**

Jenkins распоредувањето за големи тимови треба да осигура дека работните јазли ја изведуваат изградбата. Затоа, поважно е да имате големи количини на складирање на работните јазли отколку на главниот. Поставките и логовите на Jenkins стандардно се зачувуваат на главниот, така што кога планирате за големината на јачината на звукот на вашиот master, земете го предвид бројот и големината на проектите што ќе ги одржувате. Во оваа инсталација, складирањето во форма на тома на Amazon Elastic Block Store (Amazon EBS) е прикачено и на примарните и на работните јазли.

Amazon EBS обезбедува постојани количини на складирање на ниво на блок за да обезбеди ниска латентност и конзистентни перформанси за влез/излез, намалени според потребите на вашата апликација.

Обемот на EBS е дизајниран за годишна стапка на неуспех (AFR) помеѓу 0,1% - 0,2%, каде што неуспехот се однесува на целосно или делумно губење на волуменот, во зависност од големината и перформансите на волуменот. Ова ги прави звуците на EBS 20 пати посигурни од типичните стокови дискови, кои не успеваат со AFR од околу 4%.

**Тип на инстанца**

Кога ја градите околината на Jenkins на Amazon EC2, земете го предвид CPU, вмрежувањето и складирањето. Ве охрабруваме да направите репер за вашиот проект кој се базира на неколку типови примероци на Amazon EC2 со цел да се избере најсоодветната конфигурација.

Во нашите евалуации направивме споредба на пет различни типови на примери: t2.large, m3.medium и m4.large, m4.xlarge и m4.2xlarge. Секој репер симулираше сообраќај од 100 истовремени корисници кои вчитуваа повеќе страници во контролната табла на Jenkin за подолг период од 10 минути.

Генерално, откривме дека m4.large е најдобрата вредност за перформансите. Просечното искористување на процесорот при тестирање на оптоварување не надминува 3%, со просечно време на одговор од 220 милисекунди. Како што се очекуваше, големини m4.xlarge и m4.2xlarge функционираа добро, но со повисока цена на час; затоа, m4.large останува најдобриот избор за нашите потреби.

M3.medium, иако добар избор за многу апликации, не функционираше толку добро како m4.large и имаше просечна искористеност на процесорот над 80% за време на тестирањето.

T2.large се покажа добро во текот на првите неколку минути од тестирањето. Меѓутоа, бидејќи примероците t2 нудат перформанси што можат да се распрснат, постојаниот износ на голем сообраќај од 100 корисници ги исцрпи достапните кредити на процесорот, а перформансите значително се намалија. Понатамошното тестирање со помалку корисници (т.е. 10 корисници) забележа подобри резултати. Така, ако имате релативно мал тим и не очекувате честа или голема употреба од вашиот Jenkins master, Т2 family може да биде добра опција за вас.

Во оваа бела хартија, градиме Jenkins master со помош на m4.large, кој доаѓа со 2 vCPU и 8 GB меморија. Обезбеден е EBS волумен за солидна состојба (SSD) од 20 GB и е прикачен на примерот. Обемите за SSD за општа намена се стандардниот тип на јачина на звук EBS за Amazon EC2 и се поддржани од SSD дискови, што ги прави погодни за широк опсег на оптоварување на работа, вклучувајќи бази на податоци за мали до средни бази, развојни и тестирачки средини и волумен на подигање.**Инсталација**

**Master јазол**

1. Од Конзолата за управување AWS, стартувајте го примерот Amazon EC2 од Amazon Machine Image (AMI) што го има основниот оперативен систем што го сакате. Овој пример користи 64-битен AMI на Amazon Linux.
2. Изберете безбедносна група што ќе дозволи пристап до SSH, како и портата 80 или 8080 за да пристапите до контролната табла на Jenkins. Треба само да овозможите навлегување од IP адресите што сакате да им овозможите пристап до вашиот сервер. Видете **Securing Network Access to the Master Node** за повеќе детали.
3. Поврзете се со примерот преку SSH.
4. Ажурирајте ја алатката за управување со yum пакети



1. Преземете го најновиот пакет за кодови Jenkins.



1. Увезете клучна датотека од Jenkins-CI за да овозможите инсталација од пакетот.



1. Инсталирајте го Jenkins.



1. Започнете го Jenkins како услуга.



1. Конфигурирајте го Jenkins, сега кога е инсталиран и работи на вашата инстанца на Amazon EC2. Користете го интерфејсот за управување на портата 80 или 8080, или пристапувајте од далечина до серверот преку SSH. Во неговата стандардна конфигурација, верзиите на enенкинс 2.0 и подоцна заклучуваат пристап до интерфејсот за управување.

Првиот пат кога ја користите контролната табла на http: // <your-serveraddress>: 8080, ќе ви биде побарано да го отклучите Jenkins:



Како што е наведено на корисничкиот интерфејс, оваа лозинка можете да ја најдете во /var/lib/jenkins/secrets/initialAdminPassword. Вметнете ја вредноста во полето за лозинка, а потоа изберете Продолжи.

1. Скриптата за инсталација ве упатува на страницата Customize Jenkins. Изберете **Изберете додатоци за инсталирање** и изберете ги сите додатоци соодветни за вашата конкретна инсталација. За нашиот пример, осигурете се дека се инсталирани приклучокот Git (под **Управување со изворен код**) и приклучокот SSH Slaves (под **дистрибуирани градби**).

**Безбедносни размислувања**

Како минимум, влезниот сообраќај до Jenkins master треба да биде заклучен до специфичните опсези на IP адреси од кои очекувате сообраќај. Дополнително, вашата околина може дополнително да се обезбеди со користење на следниве методи.

**Enable SSL**

Сертификатите за безбедност на слоеви за заштита/транспортни слоеви (SSL/TLS) може да се користат за да се обезбедат мрежни комуникации и да се утврди идентитетот на веб -страниците преку Интернет. Можете да го направите ова лесно со преклопување на вашиот Jenkins master со балансот на товарот ELB. Во тој случај, можете да го користите менаџерот за сертификати AWS за лесно да обезбедите сертификат што ви дава шифрирани мрежни конекции и ги штити вашите податоци додека се движат низ жицата.