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**

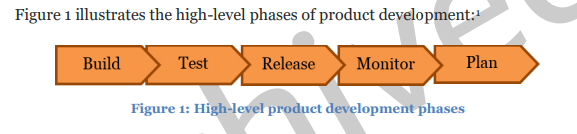
**Апстракт**

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.

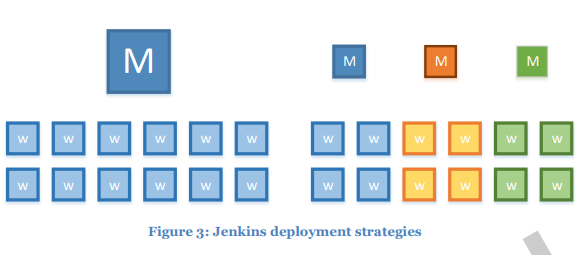
Ова е кога агентите за изградба (или јазли на работници) можат да имаат корист од инсталацијата на Jenkins со ослободување на ресурсите на главниот јазол и обезбедување прилагодени средини во кои ќе се тестираат градбите. Работнички јазол содржи агент кој комуницира со главниот сервер и работи со лесна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



**Стратегии за Jenkins Master Deployments**

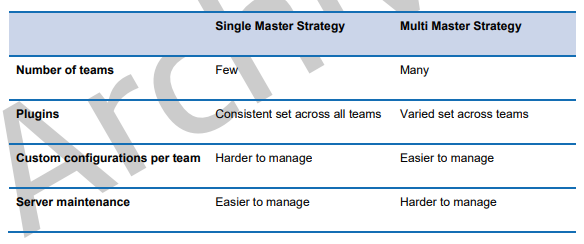
Инсталациите на Jenkins генерално спаѓаат во едно од двете сценарија:

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.

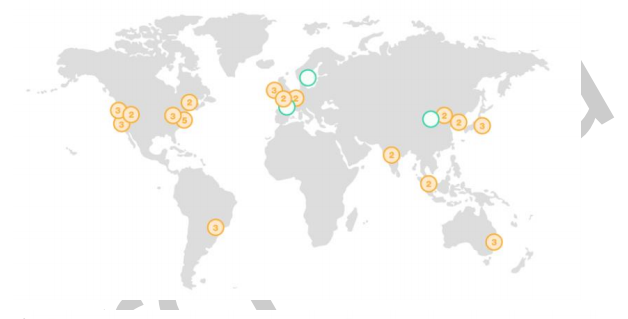
Следната табела дава некои критериуми што можете да ги користите кога избирате која стратегија најдобро одговара на вашите потреби:

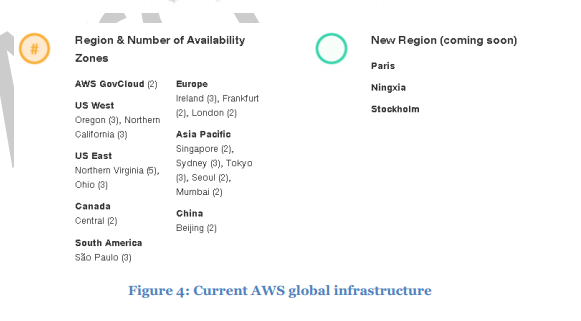


Иако постојат многу успешни стратегии за распоредување на повеќе masters на Jenkins, овој труд се фокусира на распоредување на еден master на Jenkins на Amazon EC2, но со архитектирање на тоа да биде само-заздравувачки и високо достапен. Техниките опишани овде за распоредување на еден master на Jenkins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

**Архитектура за High Availability**

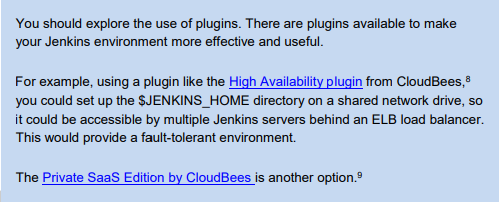
Инфраструктурата AWS (Слика 4) е изградена околу региони и зони на достапност (AZs). Регион е физичка област на светот каде AWS има повеќе Достапни зони. Достапна зона се состои од еден или повеќе дискретни центри за податоци, секој со вишок енергија, вмрежување и поврзување, сместени во посебни капацитети. 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.

**CSRF Заштита**

Фалсификат на барање меѓу страници (CSRF) е класа на напад што го принудува крајниот корисник да изврши несакани дејства врз Jenkins. Стандардно, инсталациите на Jenkins 2.x ја вклучуваат опцијата за заштита на CSRF. За да го проверите статусот на оваа поставка, изберете **Manage Jenkins**, потоа **Configure Global Security** и уверете се дека е овозможено спречување прекршочни барања за фалсификат.

**Импликација за безбедност на градење на master**

Треба да се внимава да не се изведува градење на master. Како што гледате во следните чекори за конфигурација, препорачуваме да го конфигурирате master да нема извршители и да извршувате градење само на работни јазли. Изградбите што се извршуваат на master имаат можност да читаат и менуваат датотеки во $ JENKINS\_HOME, до кои, доколку се дојде до злонамерен пристап, може да влијаат на целата инсталација на Jenkins и безбедноста на вашиот систем.

**Контрола на пристап на работниот јазол**

Верзиите на Jenkins 1.5.80.x и подоцна имаат подсистем што воспоставува бариера помеѓу работните јазли и master за безбедно да дозволат помалку доверливи работнички јазли да се поврзат со master.

Вашата инсталација Jenkins 2.x треба стандардно да ја овозможи оваа бариера. Можете да го потврдите ова со посета на „Manage Jenkins, “, потоа „Configure Global Security, “ и осигурувајќи дека е овозможено „Enable Slave -> Master Access Control “.

**Конфигурирајте автентикација на корисникот**

За време на процесот на инсталација, можете да изберете да креирате прв администратор. Создадете главен корисник што може да се користи за создавање други групи и корисници, а потоа продолжете до контролната табла на Jenkins.

Автентикацијата на корисникот може да се обезбеди преку голем број методи:

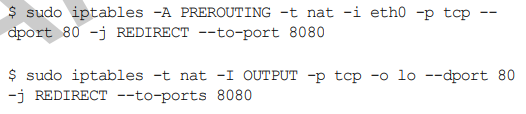
* Наједноставната шема за автентикација е да се користи базата на податоци на Jenkins. Корисничките сметки може да се креираат со помош на контролната табла на Jenkins (изберете Manage Jenkins, потоа Configure Global Security).
  + - Опцијата за безбедност базирана на Матрикс нуди најпрецизна контрола врз корисничките привилегии. Користејќи ја оваа опција, можете да наведете ситно-гранулирани дозволи за секој корисник и секоја акција што може да ја преземе.
* Ако Jenkins работи на машина со Windows, можете да го конфигурирате Jenkins да ги автентицира корисничкото име и лозинката преку Active Directory користејќи го Active Directory plugin.
* Ако Jenkins работи на Linux, истиот додаток на Active Directory може да се користи со одредување на домените на Active Directory за автентикација, или може да конфигурирате пристап до корисници/групи на Unix во вашиот локален систем. Со оваа поставка, корисниците ќе бидат најавени на Jenkins со внесување на нивното корисничко име и лозинка за Unix.
* Со приклучокот LDAP, корисниците можат да автентицираат со услуга LDAP што е во согласност со Active Directory, како што се AWS Directory Service или OpenLDAP.

**Обезбедување пристап до мрежа**

Безбедносната група делува како виртуелен заштитен ѕид кој го контролира сообраќајот кон вашите инстанци. Кога лансирате пример, поврзувате една или повеќе безбедносни групи со примерот. Бидејќи сите порти се стандардно оневозможени, додавате правила на секоја безбедносна група што овозможуваат сообраќај до или од придружните инстанци.

Кога го стартувате вашиот master, креирајте безбедносна група што овозможува навлегување во следните порти:

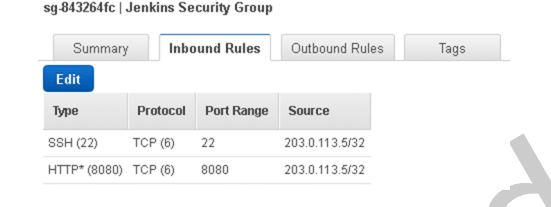
1. Порта 80 или 8080, за способност да го конфигурирате Jenkins и да комуницирате со контролната табла. Стандардно, контролната табла на Jenkins е достапна преку портата 8080, но можете да користите iptables за пренасочување на портата 80 до 8080 и да дозволите локални врски:



1. Порта 22, за да се поврзете преку SSH конекција со инстанците и да извршите одржување.

За секоја порта, ограничете го пристапот до вашата IP адреса или опсег на IP адреси користејќи го полето Source field, што го одредува сообраќајот што може да стигне до вашата инстанца. Наведете единствена IP адреса или опсег на IP адреса во белегот за рутирање меѓу-домени без класа (CIDR) (на пример, 203.0.113.5/32 како на сликата).

Ако се поврзувате од позади огнениот ѕид, ќе ви треба опсегот на IP адреси што го користат клиентите компјутери.

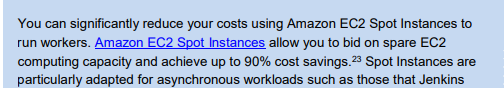


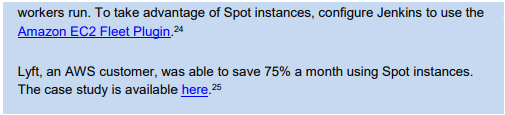
**Работнички јазли**

Работнички јазол на Jenkins е пример што испушта проекти за изградба од master. Jenkins master ја прави распределбата на задачите прилично автоматски и може да се конфигурира по проект. Корисниците кои пристапуваат до Jenkins користејќи ја контролната табла за управување можат да го следат напредокот во изградбата, да пребаруваат датотеки Javadoc и да ги гледаат и преземаат резултатите од тестовите без да треба да бидат свесни дека изградбите се направени на работни јазли.

Секој јазол на работници работи агент за работници, со што се елиминира потребата да се инсталира целосниот пакет Jenkins на тие случаи. Овие агенти може да се инсталираат со употреба на различни методи, секој со краен резултат од воспоставување двонасочна комуникација помеѓу master пример и примерот на работникот.

Откако ќе изберете тип на пример и ресурси за вашите работни јазли, најдобро е да ги изградите за општа намена, наместо да ги изградите за одреден проект. Ова е особено точно за големи тимови или во средини со различни проекти. Како по правило, запомнете дека вашите работни јазли треба да бидат спојливи - лесно се заменуваат со друг јазол во случај на дефект.





Откако ќе конфигурирате соодветен јазол за работници и ќе ги инсталирате потребните пакети и алатки (на пример, Maven или Git), креирајте AMI од кој можете брзо да лансирате други јазли.

На работните јазли ќе им треба само порта 22 за да бидат отворени за комуникација со master. Нема да им треба отворена порта 8080. Создадете безбедносна група што овозможува влез во портата 22 и ограничете го пристапот до IP адресата и на вашиот главен јазол и на вашата специфична IP адреса на вашата внатрешна мрежа.

Треба да ги измерите перформансите за вашите проекти за да го одредите соодветниот тип на инстанца и големина што ви се потребни, како и количината на простор на дискот. Во оваа бела хартија, ние наведуваме дека нашите работнички јазли на Jenkins користат тип m4.голем пример, кој доаѓа со 2 vCPU и 8 GB меморија, заедно со јачина на звук од 40 GB SSD Amazon EBS од општа намена.

Работните јазли се поврзуваат со master на различни начини - преку SSH, преку Windows преку објектот за далечинско управување (вграден во Windows 2000 или понова верзија), преку Java Web Start или преку прилагодена скрипта.

За инсталациите на Linux, SSH е најзгодниот и најпосакуваниот метод за лансирање работнички агенти и е оној што се користи во оваа бела хартија.

**Инсталација**

1. Започнете нова инстанца во истиот VPC во кој го лансиравте Jenkins master. За овој пример, го стартуваме примерот од 64-битен AMI на Amazon Linux. (Забележете дека нема да морате да го инсталирате enенкинс на вашите работници.)
2. Прикачете безбедносна група што овозможува навлегување во работните јазли преку SSH од внатрешната IP адреса на главната инсталација Jenkins.
3. Поврзете се со главниот пример преку SSH.2
4. Создадете пар клучеви за автентикација од директориумот ~/.ssh:



1. Копирајте го јавниот клуч - ќе ви треба во чекор 7



1. Од друг терминален прозорец, поврзете се со примерот на работникот преку SSH.
2. Изменете ги authorized/.ssh/овластените клучеви и додајте ја содржината на јавниот клуч од чекор 5 до крајот на датотеката. Внимавајте да не вклучите прекини во линијата што можеби сте ги копирале додека го фаќате јавниот клуч.
3. Вратете се на терминалот каде што се поврзавте со главната инстанца преку SSH и проверете дали можете да се поврзете со примерот на работникот на ист начин.



**Конфигурација**

На главниот пример, најавете се на контролната табла на Jenkins, изберете Manage Jenkins, потоа Manage Nodes и конечно New Node. За секој јазол на работници, извршете ги следниве чекори:

1. Именувајте го вашиот јазол и изберете **Permanent Agent**. Потоа изберете **ОК** за да продолжите. (Ако веќе сте го конфигурирале работниот јазол, можете да изберете **Copy Existing Node** за да ја дуплирате претходната конфигурација.)
2. Додадете опис и поставете го бројот на извршители што сакате да ги извршите на работниот јазол. План за двајца извршители по јадро.
3. Поставете **Remote Root Directory** на апсолутна патека, како/home/ec2- user/jenkins. Далечинскиот root директориум ќе ги чува податоците како што се алатката инсталација или изградба на работни простори. Ова го спречува непотребното преземање алатки или повторно проверка на изворниот код кога ќе започнат да се извршуваат конструкции на овој агент по рестартирање. Поставете ги сите други опции што ги сакате.
4. За методот на лансирање, изберете **Launch slave agents на Unix** -машините преку SSH и поставете го домаќинот на приватната IP -адреса на вашиот работнички јазол.
5. Додадете ги вашите акредитиви со избирање **Add**, потоа **корисничко име SSH со приватен клуч.**
6. Додадете ја вредноста „ec2-user“ за **Корисничко име**.
7. За приватен клуч, изберете **Enter directly**, потоа залепете ја вредноста на приватниот клуч (~/.ssh/id\_rsa) од главниот пример и изберете **Add**. Конечно, изберете **Save** за да ја зачувате конфигурацијата на јазолот на работникот.
8. Изберете го работникот од табелата со јазли, а потоа изберете **Launch agent**.

Откако ќе ги конфигурирате работните јазли, важно е да го смените бројот на извршители на 0 на master. Изберете го master од табелата со јазли и конфигурирајте ги поставките „# на извршители“ на 0. Ова гарантира дека изградбата на проекти сега ќе се случи на работните јазли, а не на master.

Тестирајте го Jenkins setup со додавање нов проект и изведување на градба

**Одвојување конфигурација од нашата Jenkins инсталација**

Во претходниот дел, ние навестивме дека опциите за високо расположливи распореди (HA) се ограничени, со исклучок на распоредувањата што ги изгради CloudBees.

Овој дел илустрира како раздвојувањето на тие конфигурациски датотеки од машината што работи со главниот јазол Jenkins овозможува зголемено ниво на сигурност и достапност. Ние ќе се потпреме на Amazon Elastic File System (Amazon EFS) за складирање на $ JENKINS\_HOME и ќе имплементираме конфигурации за лансирање на Auto Scaling за да се осигураме дека нашата инсталација може да се опорави низ Достапните зони, доколку нешто се случи со нашата инсталација Јенкинс.

**Создадете датотечен систем Amazon EFS**

1. Во AWS Management Console, одете до таблата на Amazon EFS.
2. Изберете Create File System.
3. Изберете го целниот VPC во кој сакате да го распоредите вашиот датотечен систем. Ова треба да биде истиот VPC каде што го поставивте поставувањето на Jenkins.
4. Под **Create mount targets** конфигурирајте ги целите за монтирање што ви се потребни. Инстанците се поврзуваат со датотечен систем преку цели што ги креирате. Препорачуваме да креирате цел за монтирање во секоја од Достапните зони на вашиот VPC, така што инстанците на Amazon EC2 низ вашиот VPC можат да пристапат до датотечниот систем.

При изборот на безбедносна група, осигурајте се дека Amazon EFS дозволува сообраќај од вашите инстанци на Jenkins. Овие безбедносни групи дејствуваат како виртуелен заштитен ѕид кој го контролира сообраќајот меѓу нив. Ако не обезбедите безбедносна група при креирање цел за монтирање, Amazon EFS ја поврзува стандардната безбедносна група на VPC со неа. Таа група го овозможува целиот сообраќај низ инстанци, така што ќе ја сервира нашата цел.

Без разлика, за да овозможите сообраќај помеѓу пример EC2 и цел за монтирање (а со тоа и датотечниот систем), мора да ги конфигурирате следниве правила во овие безбедносни групи:

1. Безбедносните групи што ги поврзувате со цел за монтирање мора да овозможат влезен пристап за протоколот TCP на портата 2049 за NFS од сите инстанци EC2 на кои сакате да го монтирате датотечниот систем.
2. Секој EC2 пример што го монтира датотечниот систем мора да има безбедносна група што овозможува излезен пристап до целта за монтирање на портата TCP 2049.
3. Изберете **Next Step**.
4. Додадете ознаки ако ви се потребни и изберете режим на изведба. Препорачуваме режим на изведба за општа намена за повеќето датотечни системи. Режимот на максимална I/O изведба е оптимизиран за апликации каде што десетици, стотици или илјадници инстанци EC2 пристапуваат до датотечниот систем - тој се движи до повисоки нивоа на вкупна пропусна моќ и операции во секунда со замена за малку повисоки задоцнувања за операции со датотеки.
5. Изберете **Next Step**.
6. Прегледајте ја вашата конфигурација и потоа изберете **Create File System**.

**Поставување на вашиот Jenkins Host**

1. Стартувајте пример во вашиот VPC од AMI (нашиот пример користи Amazon Linux 64-битен AMI). Следете ги инструкциите дадени порано и не заборавајте да изберете безбедносна група што овозможува SSH и HTTPS (порти 22 и 8080, соодветно).
2. Поврзете се со примерот преку SSH.
3. Ажурирајте ја алатката за управување со yum пакети:



1. Инсталирајте nfs-utils, доколку е потребно:

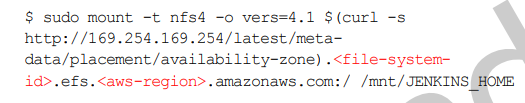


Ако го изберете Amazon Linux AMI 2016.03.0 при стартување на вашата инстанца EC2, нема да треба да инсталирате nfs-utils бидејќи веќе е вклучен во АМИ однапред

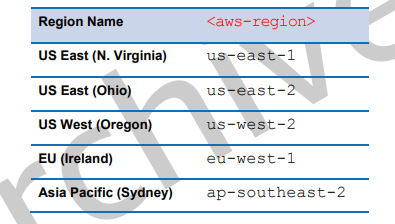
1. Создадете папка за монтирање на вашата папка $ JENKINS\_HOME:



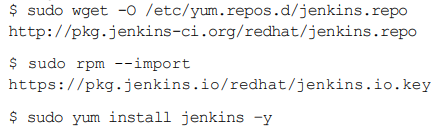
1. Монтирајте го датотечниот систем Amazon EFS на овој директориум. Користете ја следнава команда, заменувајќи ги <file-system-id> и <aws-region> заменувачите со ID на датотечен систем и регион AWS, соодветно:



Совет: <aws-region> ќе биде една од следниве вредности:



1. Инсталирај Jenkins



1. Променете ја сопственоста на новосоздадената монтажа:



1. Ажурирајте ја конфигурацијата на Jenkins:



Внатре во таа датотека, заменете ја вредноста на $ JENKINS\_HOME со /mnt /JENKINS\_HOME. Ако веќе постојат конфигурациски датотеки во постојниот директориум $ JENKINS\_HOME (стандардно//var/lib/Jenkins), проверете дали ќе ги преместите во новиот директориум/mnt/JENKINS\_HOME.

1. Стартувај го Jenkins:



1. Проверете дали споделувањето автоматски се монтира кога ќе се вклучи кутијата:



Додадете линија:



Заменете го <mount-target-DNS> со името DNS на вашиот Amazon EFS сервер. (Го користевте тоа име DNS кога го монтирате јачината на звукот EFS на Amazon на машината порано, во чекор 6.)

**Поставување група за автоматско скалирање за автоматско обновување**

Во овој момент имате сервер Jenkins кој се потпира на Amazon EFS за неговата конфигурација. Ова значи дека ако вашиот сервер Jenkins исчезне, можете да го замените со друга инсталација на Jenkins која покажува на акцијата на Amazon EFS каде се чуваат вашите податоци.

Сигурно можете рачно да го замените тој сервер (повторно да ги извршите чекорите погоре). Сепак, ние сакаме тоа да се случи автоматски, потпирајќи се на стандардните карактеристики на AWS.

1. Создадете AMI на вашата инстанца:
   * + 1. На конзолата за управување AWS, кликнете со десното копче на вашата инстанца.
       2. Изберете **Image-> Create Image**, дајте му име, изберете **Create Image** и почекајте неколку минути.
2. Создадете конфигурација за стартување:
   * + 1. На контролната табла на Amazon EC2, изберете **Launch Configuration** во панелот за навигација лево, потоа изберете **Create Auto Scaling Group**. (Конзолата може да прикаже копче **Create launch configuration**, ако веќе имате некои групи за автоматско скалирање. Наместо тоа, изберете го копчето ако тоа е случај.)
       2. Изберете **Create launch configuration**.
       3. Изберете AMI што штотуку го создадовте. (Изберете **My AMIs** во панелот за навигација за да ги прикажете вашите AMIs.)
       4. Пополнете ги потребните информации на следните неколку страници. Бидете сигурни дека ќе ја изберете безбедносната група што ја создадовте порано. (Портите 22 и 8080 треба да бидат отворени.)
       5. Конечно, изберете **Create launch configuration** од Review page и изберете клуч до кој имате пристап.
3. Создадете група за автоматско скалирање:
   * + - 1. Пред да ги извршите чекорите подолу, исклучете ја постојната, рачно креирана инстанца на Jenkins, за да не се меша со оние што ќе се вртат како дел од вашата група за автоматско скалирање.
         2. На контролната табла на Amazon EC2, изберете **Auto Scaling Group** во панелот за навигација, потоа изберете **Create Auto Scaling group**.
         3. Изберете ја конфигурацијата за стартување што ја создадовте и изберете **Next Step**.
         4. Именувајте ја вашата група за автоматско скалирање и поставете ја големината на групата = 1. (Запомнете, можеме да работиме само по еден пример во исто време.)
         5. Изберете соодветен VPC и подмрежи. (Бидете сигурни да изберете подмрежи кои се на различни Достапни зони.) Изберете **Next: Configure scaling policies.**
         6. Не користете Scaling Policies, така што групата останува во почетната големина.
         7. Кликнете на следните неколку страници и конечно изберете **Create Auto Scaling group** на Review Page.

**Проширувања**

Во овој момент, ако рачно го исклучите вашиот сервер Jenkins, Автоматското скалирање започнува нов кој ќе започне од местото каде што застана стариот сервер. Размислете исто така да го направите следново:

* Поставете балансирач на оптоварување што покажува на вашата група за автоматско скалирање за да можете постојано да го пронајдете вашиот сервер користејќи го името на DNS за балансирање на оптоварување. (Со замена на вашиот сервер, неговата IP адреса ќе се промени.)
* Како што споменавме порано, користете Amazon EFS или кој било друг дистрибуиран и високо сигурен механизам за складирање.
* Снимајте го ова поставување во образец AWS CloudFormation за да го олесните репродукцијата.

**Најдобри практики**

**Безбедност**

Во стандардната конфигурација, Jenkins не извршува никакви безбедносни проверки што би можеле да ги изложат лозинките, сертификатите и приватните податоци, како и да остави отворени работни места и конфигурации за секого. Користејќи безбедносни групи, конфигурирајте ја безбедноста на корисникот и безбедноста на мрежата и на вашиот главен пример и на работните јазли за да ги ограничите векторите на кои напаѓачот може да го компромитира вашиот систем.

**Инстанци**

Стартувајте го вашиот Jenkins master од инстанца што има доволно процесор и мрежен пропусен опсег за да се справи со истовремени корисници. Конфигурирајте ги работните јазли така што тие се идентични едни со други. Тие треба да работат на истото семејство од иста генерација, и изградбата треба да се случи на работните јазли, а не на master. Работните јазли треба да се споиви – да можат брзо да се фрлат и да се воведат или да се додадат во кластерот со што е можно помала рачна интеракција. Користете AMI како што е опишано порано за да креирате стандардна слика за вашите работни јазли. Потоа, стартувајте дополнителни работни јазли по потреба врз основа на оваа слика.

Ако вашите тимови се градат во посветени или предвидливи времиња, можете да ги запрете јазлите на работниците кога не работат работните места и да ги вклучите само кога ви се потребни. На овој начин не плаќате за неактивни случаи.

**Следење**

Во секое време, следете ги вашите случаи, особено перформансите на процесорот и дискот. Искористете ги предностите на алармите на Amazon CloudWatch за да ги следите ресурсите на пример, како што се користењето на процесорот или користењето на дискот. Предупредувањата за е -пошта и СМС може да се конфигурираат за веднаш да ве известат кога настани како што се нискиот простор на дискот или високата употреба на процесорот го надминуваат прагот што го дефинирате.

**Резервна копија и реставрација**

Одржувањето редовна резервна копија на вашиот Jenkins master е клучно за обезбедување стабилна средина. Резервната копија гарантира дека вашиот пример на Jenkins може да се врати во случај на оштетување или губење на податоци или погрешна конфигурација на Jenkins, што го остава во употреблива состојба.

Може да направите резервна копија или со сликање на целиот сервер или со правење резервна копија од директориумот $ JENKINS\_HOME.

Amazon EBS обезбедува функција за правење резервни копии на податоците за вашите томови на Amazon EBS до Amazon Simple Storage Service (S3) со преземање на слични снимки. Силно препорачуваме да правите редовни снимки од томовите на EBS на Amazon што го поддржуваат вашиот Jenkins master. Бидејќи можете да стартувате нов том врз основа на слика, можете брзо да закрепнете во случај на неуспех.

Наместо да направите слика од целиот волумен, можете да изберете само да направите резервна копија од директориумот $ JENKINS\_HOME, кој ги содржи вашите конфигурации специфични за Jenkins. Кога враќате, едноставно стартувате нов Jenkins master и го заменувате директориумот $ JENKINS\_HOME со содржината на резервната копија.

Конечно, постојат многу приклучоци за управување со резервни копии за вашиот директориум $ JENKINS\_HOME, како што е додатокот S3, кој ја поддржува вашата конфигурација на Amazon S3 - дизајниран да обезбеди 99,999999999% издржливост и 99,99% достапност.

**Понатамошно читање**

За повеќе информации за најдобрите практики за користење на enенкинс во животниот циклус на вашите проекти, ве охрабруваме да го прочитате Jenkins Best Practices wiki.

**Контејнерирано распоредување**

Контејнерите се форма на виртуелизација на оперативниот систем што ви дозволуваат да извршите апликација и нејзините зависности во процеси изолирани од ресурси.

**Преглед на технологија за контејнери**

Контејнерите ви овозможуваат лесно да ги спакувате кодот, конфигурациите и зависностите на апликацијата во шаблон наречен слика, кој се користи за стартување на контејнерот. Контејнерите можат да помогнат да се осигура дека апликациите се распоредуваат брзо, сигурно и доследно, без оглед на опкружувањето за распоредување. Исто така, имате повеќе грануларна контрола врз ресурсите, што може да ја подобри ефикасноста на вашата инфраструктура.